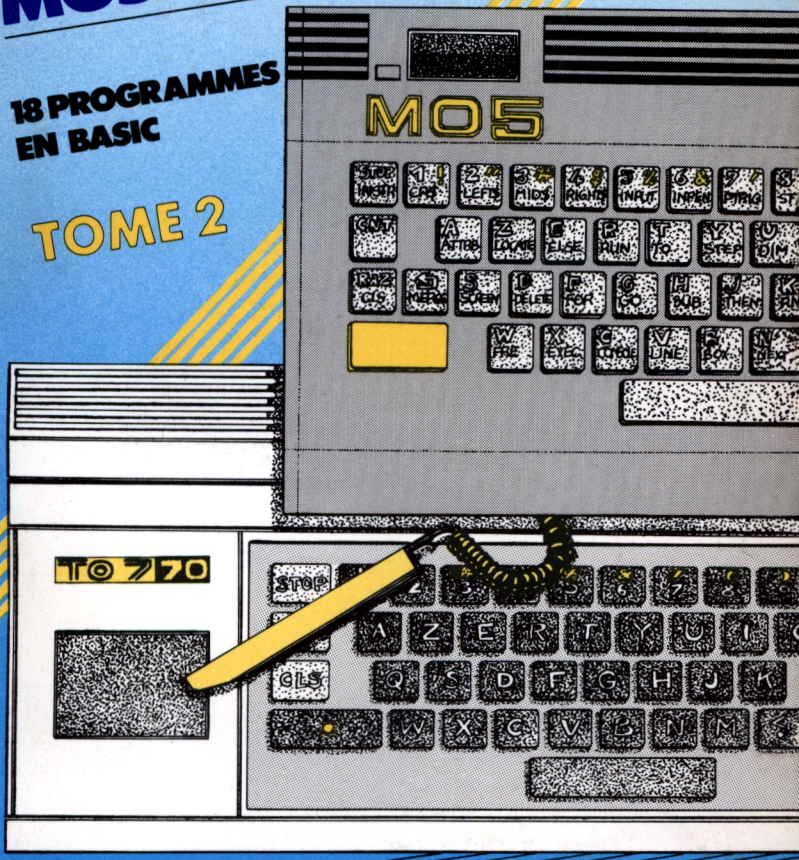
BOITE A OUTILS POUR

MO5 ET TO7/70

18 PROGRAMMES EN BASIC

TOME 2



michel martin

MegaD
POCHE

Boîte à outils pour MO5 et TO7/70

Boîte à outils. MO5 et TO7/70 — I.S.B.N. 2.87109.016.5
© s.a. SAPECA, Uccle (Belgique) 1984

Toute reproduction, non strictement réservée à l'usage du copiste, d'un extrait quelconque de ce livre par quelque procédé que ce soit, est interdite sans autorisation écrite de l'éditeur.

Les collections **MégaO** (Poche) sont éditées par SAPECA, 5 avenue de la Ferme Rose, 1180 Bruxelles (Roze Hoevelaan - 1180 Brussel). Le label **MégaO**, les titres des collections et la présentation des volumes sont déposés conformément à la loi. Distributeurs en **France :** PSI DIFFUSION, BP 86, 77402 Lagny-sur-Marne Cedex. Pour le **Canada** et les **USA :** S.C.E. Inc., 65 Avenue Hillside, Westmount - Québec H3 Z1 W1. Pour le **Benelux :** PSI Benelux, 5 Avenue de la Ferme Rose, 1180 Bruxelles.

Boîte à outils pour MO5 et TO7/70

18 programmes en Basic

TOME 2

par
Michel Martin

POCHE
1984

Avertissement

Ce livre s'adresse aux possesseurs de MO5 ou TO7 qui désirent mettre en oeuvre des programmes immédiatement opérationnels. Il constitue la suite de "Boîte à outils pour MO5/TO7 - tome 1", mais les programmes présentés ici sont d'un niveau plus complexe.

STRUCTURE DU LIVRE

Chaque programme fait l'objet de cinq parties :

- **présentation** : description macroscopique du programme ;
- **description modulaire du programme** : niveaux 1, 2 et 3 ;
- **identification des variables** : description fonctionnelle ;
- **organigramme** : définition de la structure du programme ;
- **listing du programme.**

COMMENT S'Y RETROUVER ?

Les programmes sont classés par thème (voir sommaire).

TYPE DE PROGRAMMATION

Dans la mesure du possible, les programmes sont :

- **structurés** : ils possèdent une base (ou structure) bien définie et logique ;
- **hiérarchisés** : ils sont composés de modules de niveaux de complexité (ou hiérarchie) différents.

Chaque programme possède généralement trois niveaux hiérarchiques :

- niveau 1 : module de commande (niveau hiérarchique élevé) ;
- niveau 2 : trois modules principaux : début, traitement et fin ;
- niveau 3 : procédures élémentaires, souvent sollicitées par les niveaux 1 et 2 (niveau hiérarchique bas).

COMPATIBILITE MO5/TO7

Exception faite de **"Palette de couleurs"**, tous les programmes fonctionnent indifféremment sur TO7 ou sur MO5.

En ce qui concerne "Palette de couleurs", la version proposée fonctionne sur les deux ordinateurs, mais est mal exploitée si vous possédez un MO5. En effet, TO7 possède huit couleurs alors que MO5 en possède seize.

Si vous possédez un MO5, les lignes suivantes devront être modifiées sur la version de base pour en augmenter l'intérêt :

```
60 :      FOR I=0 TO 15
2070 :    FOR J=0 TO 15
2075 :      LOCATE 2,J+6
```

Sommaire

JEUX - D'adresse ou de rapidité

- Les piranhas attaquent 10
- Atterrissage forcé 18
- Mur de briques 26
- Slalom ski 33

JEUX - De hasard

- Course de lévriers 42
- Pendu 48

JEUX - De réflexion

- Grenouilles et Crapauds 56
- Tours de Hanoï 63
- Taquin 69
- Awari 75
- Vocabulaire 83

GRAPHIQUES ET MATHEMATIQUES

- Tracé de courbes y=f(x) 92
- Tracé de courbes y(t), x(t) 97
- Tracé de courbes y(t), x(t) imbriquées 103
- Renverse 105

UTILITAIRES

- Palette de couleurs 112
- Gestion de fichiers 115
- Hard-Copy d'écran basse résolution 125

Annexe 127

JEUX D'ADRESSE OU DE RAPIDITÉ

Les piranhas attaquent

PRESENTATION

Un baigneur kamikaze (en l'occurrence, vous) se baigne dans une mare infestée de piranhas. L'écran représente la mare.

Vous pouvez vous déplacer d'une case dans n'importe quelle direction. Cela fait, chacun des piranhas se déplace d'une case dans **votre** direction.

Heureusement, il existe un moyen simple pour déjouer l'appétit cannibale de ces redoutables prédateurs : cachez-vous derrière un rocher. Le piranha ne peut freiner son élan. Il mange le rocher et meurt par indigestion.

Si vous avez eu le courage d'arriver jusqu'à cette ligne, je vous souhaite bonne chance...

ORGANIGRAMME DU JEU

Le programme a la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en trois actions élémentaires :

- attente d'une demande de déplacement du baigneur ;
- déplacement du baigneur si demande correcte ;
- déplacement de chaque piranha ;

schématisées dans l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 80** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : module d'initialisation.
lignes 2000 à 2060 : jeu.
lignes 3000 à 3150 : fin de la partie.

Niveau 3 - **lignes 1100 à 1460** : affichage des règles du jeu.
lignes 1700 à 1760 : définition des caractères graphiques.
lignes 1800 à 1990 : initialisation du jeu.
lignes 2200 à 2400 : déplacement du baigneur.
lignes 2500 à 2740 : déplacement des piranhas.

IDENTIFICATION DES VARIABLES

A : code ASCII du déplacement joueur.

A$: attente d'un caractère pour passer à la suite.

DX,DY : directions OX et OY de déplacement piranha.

FIN : indicateur de fin de partie.

I : index de boucles.

NB : nombre de piranhas.

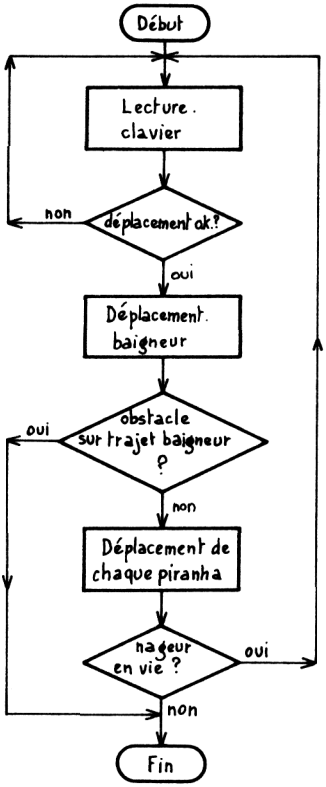
X,Y : tableaux de déplacement des piranhas.

XN,YN : position OX et OY du nageur.

XP,YP : position OX et OY des piranhas.

XR,YR : position OX et OY des rochers.

ORGANIGRAMME



```
10 :    REM LES PIRANHAS ATTAQUENT
20 :
25 :      CLEAR,,9 'Declaration de 9 motifs 9raPhi9ues
30 :      GOSUB 1000 'Presentation et Initialisation
40 :      GOSUB 2000 'Jeu
50 :      GOSUB 3000 'Fin du Jeu
60 :
65 :    END
70 :
80 :    REM ****************************************
1000 :  REM Presentation et Initialisation
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    GOSUB 1800 'Initialisation
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1145 :
1150 :      ATTRB 0,1 : COLOR 1 : CLS
1155 :      PRINT:PRINT SPC(9)"LES PIRANHAS ATTAQUENT"
1160 :
1170 :    REM Objet
1180 :
1185 :      ATTRB 0,0 : PRINT:PRINT
1190 :      COLOR 4
1195 :      PRINT"OBJET : ";
1200 :      COLOR 6
1205 :      PRINT"Jeu d'adresse et de reflexion."
1210 :
1220 :    REM Job
1225 :
1226 :      GOSUB 1700 'Definition des motifs 9raPhi9ue
s
1230 :      PRINT
1235 :      COLOR 5
1240 :      PRINT"JOB    : ";
1245 :      COLOR 6
1250 :      PRINT"Vous etes dans une mare Pleine"
1260 :      PRINT"          de Piranhas. Les * rePresente
nt"
1270 :      PRINT"          des rochers, les + des Piranh
as"
1280 :      PRINT"          et 0 le Joueur."
```

→

```
1290 :      PRINT"          Les touches ERT DFG et CVB"
1300 :      PRINT"          Permettent de se dePlacer dan
s"
1310 :      PRINT"          toutes les directions selon l
e"
1320 :      PRINT"          schema ci-dessous :":PRINT
1330 :      PRINT"          Touches ERT DePlacement "GR$(
0)GR$(1)GR$(2)
1335 :      PRINT"             ''    DFG       ''      "GR$(
3)GR$(4)GR$(5)
1340 :      PRINT"             ''    CVB       ''      "GR$(
6)GR$(7)GR$(8)
1350 :
1360 :    REM Debut
1370 :
1380 :      PRINT
1385 :      COLOR 3
1390 :      PRINT"Debut : ";
1400 :      COLOR 6
1410 :      PRINT"Pour commencer, aPPuyez sur"
1420 :      PRINT"          une touche 9uelcon9ue."
1430 :
1440 :      A=RND:A$=INKEY$:IF A$="" THEN 1440 'Attente
 de l'aPPui
1445 :
1450 :  RETURN
1460 :  REM ****************************************
1700 :  REM Confection des 9 motifs 9rabhi9ues
1705 :
1710 :    DEFGR$(0)=0,0,60,48,40,36,2,0
1715 :    DEFGR$(1)=4,14,21,4,4,4,4,0
1720 :    DEFGR$(2)=0,0,31,3,5,9,16,0
1725 :    DEFGR$(3)=0,8,16,63,16,8,0,0
1730 :    DEFGR$(4)=0,0,0,4,0,0,0,0
1735 :    DEFGR$(5)=0,4,2,63,2,4,0,0
1740 :    DEFGR$(6)=1,2,36,40,48,62,0,0
1745 :    DEFGR$(7)=0,4,4,4,4,21,14,4
1750 :    DEFGR$(8)=32,16,9,5,3,15,0,0
1755 :
1756 :  RETURN
1760 :  REM ****************************************
1800 :  REM Initialisation
1810 :
1820 :    CLS 'Effacement d'ecran
1830 :    DIM X(10),Y(10) 'Tableaux de Jeu
1840 :    FOR I=1 TO 8 'Nombre de Piranhas
1850 :      XP=INT(RND*38)+1 : YP=INT(RND*24)+1
1860 :      A=SCREEN(XP,YP)
```

```
1870 :      IF A<>32 THEN I=I-1 ELSE PSET(XP,YP)"+":X(I
)=XP:Y(I)=YP
1880 :    NEXT I
1890 :    FOR I=1 TO 30 'Nombre de rochers
1900 :      XR=INT(RND*38)+1 : YR=INT(RND*24)+1
1910 :      A=SCREEN(XR,YR)
1920 :      IF A<>32 THEN I=I-1 ELSE PSET(XR,YR)"*"
1930 :    NEXT I
1940 :    XN=INT(RND*38)+1 : YN=INT(RND*24)+1
1950 :    IF SCREEN(XN,YN)<>32 THEN 1940
1960 :    PSET(XN,YN)"0"
1970 :
1980 : RETURN
1990 : REM ****************************************
2000 : REM Jeu
2010 :
2020 :    GOSUB 2200 'Deplacement du Joueur
2030 :    GOSUB 2500 'Deplacement des Piranhas
2040 :    IF FIN<>1 THEN 2000 'Boucle Principale
2045 :
2050 : RETURN
2060 : REM ****************************************
2200 : REM Deplacement du Joueur
2210 :
2220 :    A$=INPUT$(1) 'Attente d'une action clavier
2230 :    A=ASC(A$)
2240 :    IF A=69 OR A=82 OR A=84 OR A=68 OR A=70 OR A=
71 OR A=67 OR A=86 OR A=66 THEN 2260 ELSE 2200
2250 :
2260 :    PSET(XN,YN)" " : SX=XN : SY=YN
2265 :
2270 :    IF A$="E" THEN XN=XN-1 : YN=YN-1 : GOTO 2360
2280 :    IF A$="R" THEN YN=YN-1 : GOTO 2360
2290 :    IF A$="T" THEN XN=XN+1 : YN=YN-1 : GOTO 2360
2300 :    IF A$="D" THEN XN=XN-1 : GOTO 2360
2310 :    IF A$="F" THEN 2360
2320 :    IF A$="G" THEN XN=XN+1 : GOTO 2360
2330 :    IF A$="C" THEN XN=XN-1 : YN=YN+1 : GOTO 2360
2340 :    IF A$="V" THEN YN=YN+1 : GOTO 2360
2350 :    IF A$="B" THEN XN=XN+1 : YN=YN+1
2355 :
2360 :    REM Test si le deplacement demande est Possib
le
2361 :
2365 :      IF XN=39 OR XN<1 OR YN=25 OR YN<0 THEN XN=S
X : YN=SY : PSET(XN,YN)"0" : RETURN
2366 :
2370 :    A=SCREEN(XN,YN)
```

```
2375 :    IF A<>32 THEN GOSUB 2700 'Le na9eur s'est sui
cide
2380 :    PSET(XN,YN)"0"
2385 :
2390 :  RETURN
2400 :  REM ****************************************
2500 :  REM DePlacement des Piranhas
2510 :
2520 :    FOR I=1 TO 8
2530 :      IF X(I)=99 THEN 2650
2540 :      PSET(X(I),Y(I))" "
2550 :      IF X(I)+Y(I)=0 THEN 2650
2560 :      DX=XN-X(I)
2570 :      IF DX>=1 THEN X(I)=X(I)+1
2580 :      IF DX<=-1 THEN X(I)=X(I)-1
2590 :      DY=YN-Y(I)
2600 :      IF DY>=1 THEN Y(I)=Y(I)+1
2610 :      IF DY<=-1 THEN Y(I)=Y(I)-1
2620 :      IF SCREEN(X(I),Y(I))=42 THEN GOSUB 2670 : G
OTO 2650
2625 :      IF SCREEN(X(I),Y(I))=43 THEN GOSUB 2670 : G
OTO 2650
2630 :      IF SCREEN(X(I),Y(I))=48 THEN GOSUB 2700 : G
OTO 2650
2640 :      PSET(X(I),Y(I))"+"
2650 :    NEXT I
2655 :
2660 :  RETURN
2665 :  REM ****************************************
2670 :  REM Un Piranha s'est suicide
2675 :
2680 :    PLAY"MI"
2685 :    PSET(X(I),Y(I))" "
2690 :    X(I)=99 : PF=PF+1 'Le Piranha n'existe Plus
2691 :    IF PF=8 THEN FIN=1
2695 :
2696 :  RETURN
2697 :  REM ****************************************
2700 :  REM Le na9eur s'est suicide ou a ete devore
2710 :
2715 :    PLAY"L6SIPSIPSIL24"
2720 :    FIN=1 'Fin de la Partie
2725 :
2730 :  RETURN
2740 :  REM ****************************************
3000 :  REM Fin de la Partie
3010 :
3020 :    IF PF=8 THEN 3100
```

```
3025 :
3030 :    REM Partie Perdue
3035 :
3040 :      FOR I=1 TO 5
3050 :        SCREEN 6,1,0
3060 :        SCREEN 6,0,0
3070 :      NEXT I
3080 :      CLS : PRINT"Vous voila bel et bien man9e et
 di9ere !"
3085 :      GOTO 3140
3090 :
3100 :    REM Partie 9a9nee
3110 :
3120 :      CLS:PRINT"Vous vous en sortez mais soyez pl
us"
3130 :      PRINT"Prudent la Prochaine fois."
3135 :
3140 :    LOCATE 1,1,1 'Le curseur est visible
3145 :
3150 : RETURN
```

Atterrissage forcé

PRESENTATION

Vous vous trouvez dans un avion qui doit atterrir d'urgence, suite à une fuite de carburant. En dessous de vous, s'étend la forêt vierge.

Pour pouvoir se poser, l'avion doit se ménager une piste d'atterrissage en détruisant partiellement la forêt grâce au matériel laser qu'il possède. Ce matériel comprend :

- un laser vertical, de puissance limitée, mais d'autonomie infinie ;
- un super laser vertical de grande puissance permettant cinq tirs ;
- un laser horizontal permettant quatre tirs.

A vous d'employer judicieusement ces ressources pour vous poser sans dégâts.

ORGANIGRAMME DU JEU

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en trois actions élémentaires :

- déplacement de l'avion ;
- lecture du clavier ;
- si le clavier est actionné, action clavier ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1490** : affichage du menu.
lignes 1600 à 1670 : initialisation du jeu.
lignes 2100 à 2180 : jeu.
lignes 3000 à 3080 : fin de la partie.

Niveau 3 - **lignes 1700 à 1760** : définition de l'avion.
lignes 1800 à 1895 : définition du terrain d'atterrissage.
lignes 1900 à 2020 : données de parcours.
lignes 2200 à 2290 : déplacement avion.
lignes 2300 à 2390 : commande du joueur.
lignes 2400 à 2480 : avion vers le bas.
lignes 2500 à 2580 : avion vers le haut.
lignes 2600 à 2690 : demande de tir.
lignes 2700 à 2790 : demande de bombe verticale.
lignes 2800 à 2890 : demande de laser horizontal.

IDENTIFICATION DES VARIABLES

A : définition de l'avion et du terrain.

A$: attente d'un caractère pour passer à la suite.

B$: drapeau de lancement d'une nouvelle partie.

CA : carburant restant.

FIN : indicateur de fin de jeu.

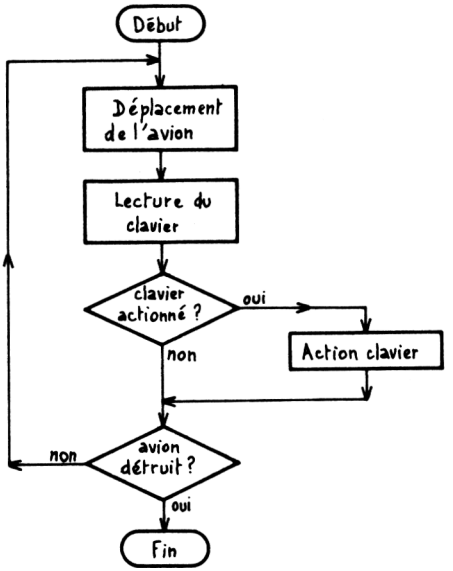
HL : laser horizontal.

I : index de boucles.

V6 : bombes verticales.

XP,YP : position de l'avion sur l'écran.

ORGANIGRAMME



```
10 :    REM ATTERRISSAGE FORCE
20 :
25 :      CLEAR,,2 'Declaration de 2 motifs 9raPhi9ues
30 :      GOSUB 1000 'Presentation
40 :      GOSUB 1600 'Initialisation
50 :      GOSUB 2100 'Jeu
60 :      GOSUB 3000 'Partie terminee
65 :      IF B$="O" THEN 40 'Une autre Partie
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation
1010 :
1020 :    SCREEN 6,0,0
1030 :
1040 :    REM Titre
1050 :
1060 :      ATTRB 1,1 : COLOR 1 : CLS
1065 :      LOCATE 1,1,0 'Le curseur disParait
1070 :      PRINT:PRINT" ATTERRISSAGE FORCE"
1080 :
1090 :    REM Objet
1100 :
1110 :      ATTRB 0,0 : PRINT
1120 :      COLOR 4
1130 :      PRINT"OBJET : ";
1140 :      COLOR 6
1150 :      PRINT"Jeu de raPidite."
1160 :
1170 :    REM Job
1180 :
1190 :      PRINT
1200 :      COLOR 5
1210 :      PRINT"JOB    : ";
1220 :      COLOR 6
1230 :      PRINT"Vous vous trouvez dans un "
1240 :      PRINT"          avion et vous devez atterrir"
1250 :      PRINT"          d'ur9ence faute de carburant.
"
1260 :      PRINT"          Malheureusement, la foret vie
r9e";
1270 :      PRINT"          s'etend au dessous de vous. P
our";
1280 :      PRINT"          Pouvoir atterrir, il faudra l
a"
1290 :      PRINT"          nettoyer. Pour cela, vous ave
z"
1300 :      PRINT"          a votre disPosition 50 bombes
 "
```

→

```
1310 :      PRINT"         4 lasers verticaux et 5"
1320 :      PRINT"         horizontaux actionnables par"
1330 :      PRINT"         les touches SPACE, C et X. Le
s"
1340 :      PRINT"         touches-fleches vers le bas e
t"
1350 :      PRINT"         vers le haut dirigent l'avion
."
1360 :
1370 :    REM Debut
1380 :
1390 :      PRINT
1400 :      COLOR 3
1410 :      PRINT"DEBUT : ";
1420 :      COLOR 6
1430 :      PRINT"Pour commencer, appuyez sur"
1440 :      PRINT"          une touche quelconque."
1460 :      A=RND : A$=INKEY$ : IF A$="" THEN 1460 'Att
ente de l'appui
1470 :
1480 : RETURN
1490 : REM ****************************************
1600 : REM Initialisation
1610 :
1620 :    GOSUB 1700 'Definition de l'avion
1630 :    GOSUB 1800 'Tirage aleatoire de la foret
1640 :    GOSUB 1900 'Affichage des donnees de vol
1650 :
1660 : RETURN
1670 : REM ****************************************
1700 : REM Definition de l'avion
1710 :
1720 :    DEFGR$(0)=0,0,0,0,0,32,49,63,63
1730 :    DEFGR$(1)=0,0,0,0,0,192,248,248
1740 :
1750 : RETURN
1760 : REM ****************************************
1800 : REM Tirage aleatoire et affichage de la foret
1810 :
1820 :    CLS
1830 :    FOR I=1 TO 30
1840 :      A=INT(RND*9)+1 'Hauteur du terrain
1850 :      FOR J=1 TO A
1860 :        PSET(I+3,22-J)CHR$(127)
1870 :      NEXT J
1880 :    NEXT I
1885 :
1890 : RETURN
```

```
1895 : REM ****************************************
1900 : REM Donnees de vol
1910 :
1920 :    CA=1000 'Carburant disPonible
1930 :    VG=4    'Bombes verticales
1940 :    HL=5    'Laser horizontal
1950 :
1960 :    LOCATE 2,22,0
1970 :    COLOR 3
1980 :    PRINT"Carburant : "CA" Bombes vert. :"VG
1990 :    LOCATE 21,23 : PRINT"Laser horiz. :";HL
2000 :
2010 :
2020 : REM ****************************************
2100 : REM Jeu
2110 :
2120 :    GOSUB 2200 'DePlacement de l'avion
2130 :    A$=INKEY$
2140 :    IF A$<>"" THEN GOSUB 2300 'Commande du Joueur
2150 :    IF FIN<>1 THEN 2100 'Boucle PrinciPale
2160 :
2170 : RETURN
2180 : REM ****************************************
2200 : REM DePlacement de l'avion
2210 :
2220 :    PSET(XP,YP)" " : PSET(XP+1,YP)" " 'Effacement
 de l'avion
2230 :    IF XP=33 THEN XP=2 ELSE XP=XP+1 'Position hor
izontale
2240 :    IF SCREEN(XP+1,YP)<>32 THEN FIN=1 'Collision
2250 :    PSET(XP,YP)GR$(0) : PSET(XP+1,YP)GR$(1) 'Avio
n
2260 :    IF XP=32 AND YP=20 THEN G=1:FIN=1 'Partie ga9
nee
2270 :
2280 : RETURN
2290 : REM ****************************************
2300 : REM Commande du Joueur
2310 :
2320 :    IF ASC(A$)=10 THEN 2400 'DePlacement vers le
bas
2330 :    IF ASC(A$)=11 THEN 2500 'DePlacement vers le
haut
2340 :    IF A$=" " THEN 2600 'Demande de tir
2350 :    IF A$="C" AND VG<>0 THEN 2700 'Demande de bom
be verticale
2360 :    IF A$="X" AND HL<>0 THEN 2800 'Demande de las
er horizontal
```

```
2370 :
2380 : RETURN
2390 : REM ****************************************
2400 : REM Deplacement de l'avion vers le bas
2410 :
2420 :    IF YP>20 THEN RETURN
2430 :    PSET(XP,YP)" " : PSET(XP+1,YP)" " 'Effacement
 de l'avion
2440 :    YP=YP+1
2450 :    IF SCREEN(XP,YP)<>32 OR SCREEN(XP+1,YP)<>32 T
HEN FIN=1 'Collision
2460 :    PSET(XP,YP)GR$(0) : PSET (XP+1,YP)GR$(1) 'Avi
on
2470 :
2475 : RETURN
2480 : REM ****************************************
2500 : REM Deplacement de l'avion vers le haut
2510 :
2520 :    IF YP<1 THEN RETURN
2530 :    PSET(XP,YP)" " : PSET(XP+1,YP)" " 'Effacement
 de l'avion
2540 :    YP=YP-1 'L'avion monte
2550 :    PSET(XP,YP)GR$(0) : PSET(XP+1,YP)GR$(1) 'Avio
n
2560 :
2570 : RETURN
2580 : REM ****************************************
2600 : REM Demande de tir
2610 :
2620 :    PLAY"DO" 'Indique que la commande est prise e
n compte
2630 :    FOR I=1 TO 8
2640 :      IF YP+I<21 THEN PSET(XP+1,YP+I)" "
2650 :    NEXT I
2660 :    CA=CA-30 : IF CA<0 THEN FIN=1 'Plus de carbur
ant
2670 :    LOCATE 14,22 : PRINT CA
2680 :
2685 : RETURN
2690 : REM ****************************************
2700 : REM Demande de bombe verticale
2710 :
2720 :    PLAY"L6DOPDOL24" 'Indique que la commande est
 acceptee
2730 :    FOR I=1 TO 8
2740 :      IF YP+I<21 THEN PSET(XP+1,YP+I)" " : PSET(X
P,YP+I)" "
2750 :    NEXT I
```

```
2760 :    CA=CA-80 : VG=VG-1 : IF CA<1 THEN FIN=1 'Plus
 de carburant
2770 :    LOCATE 14,22 : PRINT CA
2775 :    LOCATE 35,22 : PRINT VG
2780 :
2785 : RETURN
2790 : REM ****************************************
2800 : REM Demande de laser horizontal
2810 :
2820 :    PLAY"L6MIPMIL24" 'Indique que la commande est
 acceptee
2825 :    IF HL=0 THEN RETURN
2830 :    FOR I=2 TO 7
2835 :      IF XP<30 THEN PSET(XP+I,YP)" "
2840 :    NEXT I
2850 :    CA=CA-50 : HL=HL-1 : IF CA<0 THEN FIN=1 'Plus
 de carburant
2860 :    LOCATE 14,22 : PRINT CA
2870 :    LOCATE 35,23 : PRINT HL
2880 :
2885 : RETURN
2890 : REM ****************************************
3000 : REM Fin de Partie
3010 :
3020 :    PLAY"L3SIPSIPSIPSIPSIL24"
3030 :    CLS : COLOR 6
3040 :    IF G=1 THEN PRINT"Pas mal." ELSE PRINT"Ca dev
ait arriver !"
3050 :    LOCATE 0,6
3060 :    INPUT"Une autre Partie (O/N)";B$
3070 :    IF B$="O" THEN FIN=0 : XP=0 : YP=0
3075 :
3080 : RETURN
```

Mur de briques

PRESENTATION

Un mur de briques de quatre lignes (quarante briques par ligne) occupe la partie haute de l'écran. Le but du jeu est de détruire toutes les briques en faisant rebondir une balle sur une raquette dirigée par les touches-flèches droite et gauche. Quand la balle touche une brique, cette dernière disparaît et la balle rebondit.

Le joueur peut perdre trois fois la balle avant que la partie ne s'arrête. La partie est commentée après la perte des trois balles allouées, ou après destruction de toutes les briques.

ORGANIGRAMME DU JEU

Le programme a la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- déplacement de la balle ;
- déplacement de la raquette si le clavier est actionné ;
- test fin de jeu ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 80** : module de commande.

Niveau 2 - **lignes 1000 à 1430** : présentation du jeu.
lignes 2000 à 2070 : initialisation du jeu et jeu.
lignes 5000 à 5080 : fin du jeu.

Niveau 3 - **lignes 2100 à 2530** : initialisation du jeu.
lignes 3000 à 3060 : jeu.
lignes 3100 à 3200 : action sur la balle.
lignes 3300 à 3360 : renvoi de la balle.
lignes 3400 à 3490 : démolition d'une brique.
lignes 3500 à 3550 : balle perdue.
lignes 3600 à 3660 : action du joueur.
lignes 3700 à 3760 : raquette vers la gauche.
lignes 3800 à 3860 : raquette vers la droite.
lignes 4000 à 4100 : fin de la balle.

IDENTIFICATION DES VARIABLES

A$: attente d'un caractère pour passer à la suite.

BA : nombre de balles utilisées.

B$: code de déplacement raquette.

FIN : indicateur de fin de partie.

I,J : index de boucles.

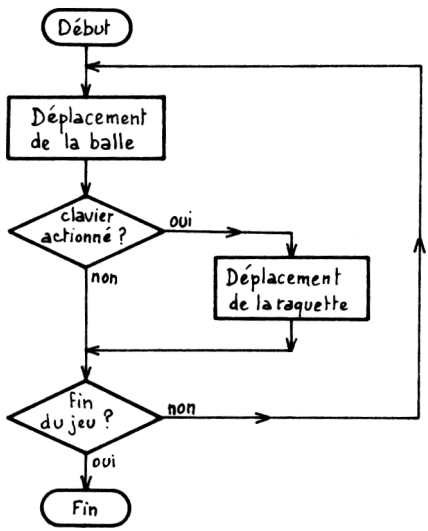
SC : score du joueur.

XB,YB : position de la balle.

XR,YR : position de la raquette.

VX,VY : vecteur de déplacement de la balle.

ORGANIGRAMME



```
10 :    REM Mur de briques
20 :
25 :      CLEAR ,,2 'Declaration de deux motifs graphiq
ues
30 :      GOSUB 1000 'Presentation
40 :      GOSUB 2000 'Initialisation et Jeu
50 :      GOSUB 5000 'Fin du jeu
60 :
70 :    END
80 :    REM ****************************************
1000 :  REM Presentation du jeu
1010 :
1020 :    SCREEN 6,0,0 : LOCATE 1,1,0
1030 :
1040 :    REM Titre
1050 :
1060 :      ATTRB 1,1 : COLOR 1 : CLS
1070 :      PRINT:PRINT"    MUR DE BRIQUES"
1080 :
1090 :    REM Objet
1100 :
1110 :      ATTRB 0,0 : PRINT:PRINT:PRINT
1120 :      COLOR 4
1130 :      PRINT"OBJET : ";
1140 :      COLOR 6
1150 :      PRINT"Jeu de reflexe."
1160 :
1170 :    REM Job
1180 :
1190 :      PRINT:PRINT
1200 :      COLOR 5
1210 :      PRINT"JOB     : ";
1220 :      COLOR 6
1230 :      PRINT"Un mur de briques occupe la"
1240 :      PRINT"          partie haute de l'ecran."
1250 :      PRINT"          Une raquette peut se deplacer
"
1260 :      PRINT"          a gauche et a droite grace au
x"
1270 :      PRINT"          touches-fleches."
1280 :      PRINT"          Le but du jeu est de detruire
"
1290 :      PRINT"          toutes les briques en faisant
"
1300 :      PRINT"          rebondir la balle sur la"
1310 :      PRINT"          raquette."
1320 :
1330 :    REM Debut
```

```
1340 :
1350 :      PRINT:PRINT
1360 :      COLOR 3
1370 :      PRINT"DEBUT : ";
1380 :      COLOR 6
1390 :      PRINT"Pour commencer, appuyez sur"
1400 :      PRINT"          une touche quelconque."
1410 :      A=RND:A$=INKEY$:IF A$="" THEN 1410 'Attente
 de l'appui
1415 :
1420 : RETURN
1430 : REM ****************************************
2000 : REM Initialisation du jeu et Jeu
2010 :
2020 :   GOSUB 2100 'Initialisation du jeu
2030 :   GOSUB 3000 'Jeu
2040 :   GOSUB 4000 'Balle suivante
2050 :   IF FIN=0 THEN 2030
2055 :
2060 : RETURN
2070 : REM ****************************************
2100 : REM Initialisation du jeu
2110 :
2120 :   PLAY"L3"
2130 :
2140 :   REM Definition des motifs brique et raquette.
2150 :
2160 :     DEFGR$(0)=0,0,42,20,42,20,42,0
2170 :
2180 :     DEFGR$(1)=0,0,0,0,0,255,255,0
2190 :
2200 :   REM Dessin des briques
2210 :
2220 :     CLS
2230 :     FOR I=1 TO 4
2240 :       COLOR I
2250 :       FOR J=1 TO 40
2260 :         PRINT GR$(0);
2270 :       NEXT J
2280 :     NEXT I
2290 :
2300 :   REM Dessin de la raquette
2310 :
2320 :     XR=1 : YR=20 'Position initiale
2330 :     COLOR 6 : LOCATE XR,YR
2340 :     PRINT GR$(1)GR$(1)GR$(1)
2350 :
2360 :   REM Balle au depart
```

```
2370 :
2380 :      XB=3 : YB=19 'Position initiale
2390 :      VX=INT(RND*3)-1:VY=-1 'Vecteur de direction
 de la balle
2400 :
2410 :    REM Debut de la Partie et affichage du score
2420 :
2430 :      COLOR 1 : LOCATE 5,22
2440 :      PRINT"Balle 1" : BA=1
2450 :
2460 :      LOCATE 0,23 : COLOR 3
2470 :      PRINT"APPuyez sur une touche Pour commencer
."
2480 :      A$=INPUT$(1) 'Attente de l'aPPui
2490 :      LOCATE 0,23 : COLOR 6
2500 :      PRINT SPC(79)
2510 :
2520 : RETURN
2530 : REM ****************************************
3000 : REM Jeu
3010 :
3020 :    GOSUB 3100 'Action sur la balle
3030 :    B$=INKEY$ : IF B$<>"" THEN GOSUB 3600 'Action
 Joueur
3040 :    IF FIN=0 THEN 3000 'Boucle PrinciPale
3045 :
3050 : RETURN
3060 : REM ****************************************
3100 : REM Action sur la balle
3110 :
3120 :    PSET(XB,YB)" "
3130 :    IF YB+VY < 0 THEN VY=1:VX=INT(RND*3)-1 'Haut
de l'ecran
3140 :    IF XB+VX < 0 THEN VX=1 : PLAY"SI"
3150 :    IF XB+VX = 41 THEN VX=-1 : PLAY"SI"
3160 :    IF SCREEN(XB+VX,YB+VY)=0 AND YB<10 THEN GOSUB
 3400 'Demolition briQue
3170 :    IF SCREEN(XB+VX,YB+VY)=0 AND YB>10 THEN GOSUB
 3300 'Renvoi de la balle
3180 :    IF YB+VY > 20 THEN GOSUB 3500 'Perte de la ba
lle
3190 :    IF FIN=1 THEN RETURN
3191 :    IF XB+VX < 1 THEN VX=1 : PLAY"SI"
3192 :    IF XB+VY=39 THEN VX=-1 : PLAY"SI"
3193 :    XB=XB+VX:YB=YB+VY:PSET(XB,YB)"o"
3194 :
3195 : RETURN
3200 : REM ****************************************
```

```
3300 : REM Renvoi de la balle
3310 :
3320 :    VY=-1:VX=INT(RND*3)-1
3330 :    PLAY"LAL3"
3340 :
3350 : RETURN
3360 : REM ******************************************
3400 : REM Demolition brique
3410 :
3420 :    PSET(XB+VX,YB+VY)" "
3430 :    SC=SC+10*(4-YB-VY)
3435 :    LOCATE 25,22 : COLOR 4 : PRINT "Score :"SC :
COLOR 6
3440 :    VY=1 : VX=INT(RND*3)-1
3450 :    IF SC>1300 THEN FIN=1 'La Partie est gagnee
3460 :    PLAY"SIPSI"
3470 :
3480 : RETURN
3490 : REM ******************************************
3500 : REM Balle Perdue
3510 :
3520 :    PLAY"SIPSIPSIPSI" : FIN=1
3530 :
3540 : RETURN
3550 : REM ******************************************
3600 : REM Action du joueur
3610 :
3620 :    IF ASC(B$)=8 THEN GOSUB 3700 'Deplacement ver
s la gauche
3630 :    IF ASC(B$)=9 THEN GOSUB 3800 'Deplacement ver
s la droite
3640 :
3650 : RETURN
3660 : REM ******************************************
3700 : REM Deplacement de la raquette vers la gauche
3710 :
3720 :    IF XR=1 THEN RETURN 'Deplacement reffuse
3730 :    LOCATE XR+1,YR : PRINT"  "
3735 :    XR=XR-2 : LOCATE XR,YR : PRINT GR$(1)GR$(1)
3740 :
3750 : RETURN
3760 : REM ******************************************
3800 : REM Deplacement de la raquette vers la droite
3810 :
3820 :    IF XR=35 THEN RETURN 'Deplacement reffuse
3830 :    LOCATE XR-1,YR : PRINT"  "
3835 :    XR=XR+2 : LOCATE XR,YR : PRINT GR$(1)GR$(1)
3840 :
```

```
3850 : RETURN
3860 : REM ****************************************
4000 : REM Balle suivante
4010 :
4020 :   BA=BA+1 : IF BA=4 THEN FIN=1 : GOTO 4090 ELSE
 FIN=0
4030 :   LOCATE 10,22 : COLOR 1 : PRINTBA
4040 :   COLOR 3 : PRINT"Appuyez sur une touche pour c
ommencer."
4050 :   A$=INPUT$(1) 'Attente de l'appui
4055 :   LOCATE 0,23 : PRINT SPC(79)
4060 :   XB=3 : YB=19 : VY=-1
4070 :   COLOR 6
4080 :
4090 : RETURN
4100 : REM ****************************************
5000 : REM Fin du jeu
5010 :
5020 :   CLS
5030 :   IF SC>1000 THEN PRINT"Pas trop mal" ELSE PRIN
T"Bof, bof ..."
5040 :   IF BA=1 OR BA=2 THEN PRINT"Bravo !"
5050 :   LOCATE 0,5,1 'Restitution du curseur
5060 :
5070 : RETURN
5080 : REM ****************************************
```

Slalom ski

PRESENTATION

Les touches-flèches droite et gauche permettent de guider, en rotation, un skieur animé d'un mouvement de translation rectiligne uniforme. Le but du jeu est de passer entre les barrières d'un slalom inventé par MO5.

Skieurs, à vos claviers...

ORGANIGRAMME DU JEU

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- avance du skieur ;
- si le clavier est actionné, rotation du skieur ;
- test si fin de parcours ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et initialisation.
lignes 2000 à 2060 : initialisation du jeu et jeu.
lignes 4000 à 4140 : fin de la partie.

Niveau 3 - **lignes 1100 à 1510** : présentation du jeu.
lignes 1600 à 1990 : initialisation graphique.
lignes 2100 à 2420 : initialisation du jeu.
lignes 2600 à 2710 : avance skieur.
lignes 2800 à 2860 : changement de direction.

lignes 2900 à 2950 : sens trigo.
lignes 3000 à 3050 : sens inverse trigo.
lignes 3100 à 3170 : fin de changement de direction.

IDENTIFICATION DES VARIABLES

A$: attente d'un caractère pour passer à la suite.

B$: code action clavier.

I : index de boucles.

FIN : indicateur de fin de partie

NB : direction du skieur.

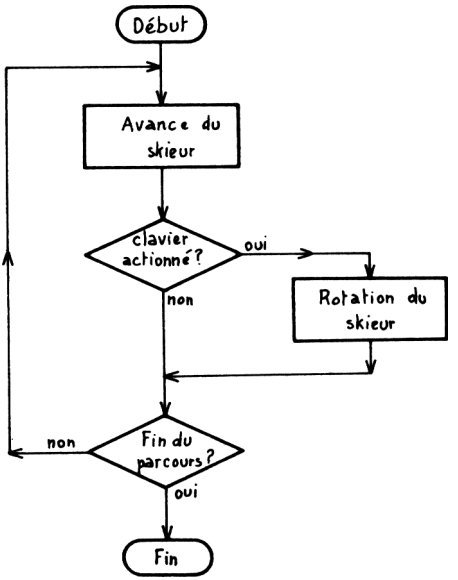
S : indicateur de sortie de piste.

VX, VY : vecteurs de changement de direction.

X : position d'un passage.

X1, X2, Y1, Y2 : coordonnées de la position du skieur.

ORGANIGRAMME



```
10 :    REM SLALOM SKI
20 :
25 :      CLEAR,,17 'Declaration de 17 motifs 9raPhique
s
30 :      GOSUB 1000 'Presentation et Initialisation 9r
aPhique
40 :      GOSUB 2000 'Initialisation du Jeu et Jeu
50 :      GOSUB 4000 'Fin de Jeu
60 :      IF R$="O" THEN 40 'Une autre Partie est en9a9
ee
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation et Initialisation 9raPhique
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    GOSUB 1600 'Initialisation 9raPhique
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation du Jeu
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1150 :
1160 :      ATTRB 1,1 : COLOR 1 : CLS
1170 :      PRINT:PRINT"     SLALOM SKI"
1180 :
1190 :    REM Objet
1200 :
1210 :      ATTRB 0,0 : PRINT:PRINT:PRINT
1220 :      COLOR 4
1230 :      PRINT"OBJET : ";
1240 :      COLOR 6
1250 :      PRINT"Jeu de raPidite."
1260 :
1270 :    REM Job
1280 :
1290 :      PRINT:PRINT:PRINT
1300 :      COLOR 5
1310 :      PRINT"JOB    : ";
1320 :      COLOR 6
1330 :      PRINT"Vous vous trouvez sur une"
1340 :      PRINT"          Piste de ski et vous devez"
1350 :      PRINT"          effectuer un slalom en vous"
1360 :      PRINT"          servant des touches T et I"
1370 :      PRINT"          Pour effectuer une rotation d
ans";
```

```
1380 :      PRINT"         le sens Tri9o ou Inverse Tri9
o."
1390 :
1400 :    REM Debut
1410 :
1420 :      PRINT:PRINT:PRINT
1430 :      COLOR 3
1440 :      PRINT"DEBUT : ";
1450 :      COLOR 6
1460 :      PRINT"Pour commencer, aPPuyez sur"
1470 :      PRINT"         une touche 9uelcon9ue."
1480 :      A$=INPUT$(1) 'Attente de l'aPPui
1490 :
1500 : RETURN
1510 : REM ****************************************
1600 : REM Initialisation 9raPhi9ue
1610 :
1620 :    REM Differentes Positions du skieur
1630 :
1640 :      DEFGR$(0)=3,3,1,29,19,17,17,19
1650 :      DEFGR$(1)=128,128,0,0,0,192,48,16
1660 :      DEFGR$(2)=4,31,0,1,0,0,0,0
1670 :      DEFGR$(3)=144,208,80,252,16,0,0,0
1680 :      DEFGR$(4)=3,3,1,1,9,7,27,51
1690 :      DEFGR$(5)=128,128,16,112,144,16,16,0
1700 :      DEFGR$(6)=20,24,20,2,1,0,0,0
1710 :      DEFGR$(7)=128,64,32,0,0,0,0,0
1720 :      DEFGR$(8)=3,3,1,1,31,17,21,22
1730 :      DEFGR$(9)=128,128,0,0,232,16,80,208
1740 :      DEFGR$(10)=20,4,4,4,0,0,0,0
1750 :      DEFGR$(11)=80,64,64,64,0,0,0,0
1760 :      DEFGR$(12)=3,3,49,45,35,35,35,7
1770 :      DEFGR$(13)=128,128,0,0,0,192,48,48
1780 :      DEFGR$(14)=8,16,32,1,2,0,0,0
1790 :      DEFGR$(15)=208,144,144,0,0,0,0,0
1800 :
1810 :    REM Passa9es
1820 :
1830 :      DEFGR$(16)=32,48,40,36,62,32,32,32
1840 :
1850 :    REM Directions en X et Y
1860 :
1870 :      FOR I=1 TO 5
1880 :        READ DX(I)
1890 :      NEXT I
1900 :
1910 :      FOR I=1 TO 5
1920 :        READ DY(I)
```

```
1930 :      NEXT I
1940 :
1950 :      DATA 1,1,0,-1,-1
1960 :      DATA 0,1,1,1,0
1970 :
1980 : RETURN
1990 : REM ****************************************
2000 : REM Initialisation du Jeu et Jeu
2010 :
2020 :    GOSUB 2100 'Initialisation du jeu
2030 :    GOSUB 2500 'Jeu
2040 :
2050 : RETURN
2060 : REM ****************************************
2100 : REM Initialisation du Jeu
2110 :
2120 :    REM Tira9e aleatoire des Positions des Passa9
es
2130 :
2140 :      FOR I=1 TO 5
2150 :        X=INT(RND*24)+9 : TX(I)=X
2160 :      NEXT I
2170 :
2180 :    REM Afficha9e du Parcours
2190 :
2200 :      CLS
2210 :      FOR I=1 TO 5
2220 :        PSET(TX(I),(I-1)*5+1)GR$(16)
2225 :        PSET(TX(I),(I-1)*5+4)GR$(16)
2230 :      NEXT I
2240 :
2250 :    REM Joueur au dePart
2260 :
2270 :      X1=2 : Y1=2 : X2=3 : Y2=3 :VX=1 : VY=0
2275 :      NB=1 : C$=GR$(0) : D$=GR$(1) : E$=GR$(2) :
F$=GR$(3)
2280 :      PSET(X1,Y1)GR$(0) : PSET(X2,Y1)GR$(1)
2285 :      PSET(X1,Y2)GR$(2) : PSET(X2,Y2)GR$(3)
2290 :
2300 :    REM Commentaires
2310 :
2320 :      COLOR 3
2330 :      LOCATE 15,0 : PRINT"SLALOM SKI"
2340 :
2350 :      COLOR 2
2360 :      LOCATE 0,22
2370 :      PRINT"Pour commencer, aPPuyez sur une touch
e"
```

```
2380 :      A$=INPUT$(1) 'Attente de l'aPPui
2390 :
2400 :      LOCATE 0,22
2405 :      PRINT SPC(39)
2406 :      PLAY"L3"
2410 :
2415 : RETURN
2420 : REM ****************************************
2500 : REM Jeu
2510 :
2520 :    GOSUB 2600 'Avance skieur
2530 :    B$=INKEY$:IF B$<>"" THEN GOSUB 2800 'Chan9eme
nt de direction
2540 :    IF FIN<>1 THEN 2500 'Boucle PrinciPale de Jeu
2550 :
2560 : RETURN
2570 : REM ****************************************
2600 : REM Avance du skieur
2610 :
2620 :    PSET(X1,Y1)" " : PSET(X2,Y1)" "
2625 :    PSET(X1,Y2)" " : PSET(X2,Y2)" "
2630 :    X1=X1+VX : X2=X2+VX : Y1=Y1+VY : Y2=Y2+VY
2640 :    IF X1=0 OR X1=36 OR Y1=0 OR Y1=25 THEN FIN=1
: S=1 : GOTO 2700
2650 :    IF SCREEN(X1,Y1-1)=0 AND SCREEN(X1,Y1+2)=0 TH
EN GA=GA+1 : PLAY"SIPSI"
2660 :    IF X1=34 AND Y1=22 THEN FIN=1 : S=0 'Partie f
inie
2670 :    IF SCREEN(X1,Y1)=0 OR SCREEN(X1,Y2)=0 THEN PL
AY"MIPMIPMIPMIPMI"
2680 :    IF SCREEN(X2,Y1)=0 OR SCREEN(X2,Y2)=0 THEN PL
AY"MIPMIPMIPMIPMI"
2685 :     PSET(X1,Y1)C$ : PSET(X2,Y1)D$
2686 :     PSET(X1,Y2)E$ : PSET(X2,Y2)F$
2690 :
2700 : RETURN
2710 : REM ****************************************
2800 : REM Chan9ement de direction
2810 :
2820 :    IF B$="T" THEN GOSUB 2900 'Sens Tri9o
2830 :    IF B$="I" THEN GOSUB 3000 'Sens Inverse Tri9o
2840 :
2850 : RETURN
2860 : REM ****************************************
2900 : REM Sens Tri9o
2910 :
2920 :    IF NB=1 THEN RETURN
2930 :   NB=NB-1 : GOSUB 3100
```

```
2940 :
2945 : RETURN
2950 : REM ****************************************
3000 : REM Sens Inverse Tri9o
3010 :
3020 :    IF NB=5 THEN RETURN
3030 :    NB=NB+1 : GOSUB 3100
3040 :
3045 : RETURN
3050 : REM ****************************************
3100 : REM Fin de chan9ement de direction
3110 :
3120 :    VX=DX(NB) : VY=DY(NB)
3130 :    IN=(NB-1)*4 : IF NB=5 THEN IN=0
3140 :    C$=GR$(IN) : D$=GR$(IN+1) : E$=GR$(IN+2) : F$
=GR$(IN+3)
3150 :
3160 : RETURN
3170 : REM ****************************************
4000 : REM La Partie est terminee
4010 :
4020 :    CLS : COLOR 6
4030 :    IF S<>1 THEN 4050
4040 :    PRINT"alors comme ca, on sort des Pistes !!"
: GOTO 4090
4050 :    PRINT:PRINT:PRINT
4060 :    PRINT"Parcours effectue avec "5-GA" faute(s).
"
4070 :    IF GA=5 THEN PRINT"Pas mal." ELSE PRINT"Minab
le ..."
4080 :
4090 :    LOCATE 0,12,1
4100 :    INPUT"Une autre Partie (O/N)";R$
4110 :    S=0 : FIN=0 : GA=0 : LOCATE 0,14,0
4120 :
4130 : RETURN
4140 : REM ****************************************
```

JEUX DE HASARD

Course de lévriers

PRESENTATION

Ce jeu de hasard anime quatre levriers qui se disputent la première place dans une course de vitesse. Le déplacement d'un lévrier étant aléatoire, il est très difficile de trouver leur ordre d'arrivée. Ce programme n'a donc qu'une valeur symbolique (animation).

ORGANIGRAMME DU JEU

Le programme a la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- déplacement de chaque chien de Ø ou 1 case, selon un rythme aléatoire ;
- test si les quatre chiens sont arrivés ;

d'où l'organigramme ci-contre.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et initialisation.
lignes 2000 à 2060 : jeu.
lignes 3000 à 3160 : arrivée.

Niveau 3 - **lignes 1100 à 1520** : présentation.
lignes 1600 à 1710 : initialisation des constantes et variables.
lignes 2100 à 2570 : initialisation du jeu.
lignes 2600 à 2790 : course.

IDENTIFICATION DES VARIABLES

A : tirage du déplacement d'un chien.

A$: attente d'un caractère pour passer à la suite.

D1 : argent en caisse.

GA : nombre de chiens arrivés dans l'ordre prévu.

GN : somme gagnée pour la partie.

I,J : index de boucles.

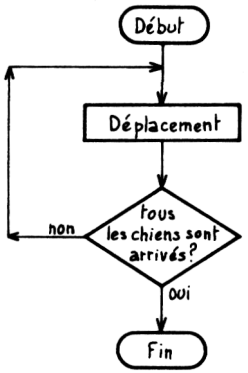
P : position des chiens sur la piste.

R$: validation d'une nouvelle partie.

T : tableau donnant la position des chiens (repos ou galop).

X : tableau d'ordre d'arrivée prévu.

ORGANIGRAMME



```
10 :    REM COURSE DE LEVRIERS
20 :
25 :      CLEAR,,8 'Definition de 8 motifs 9raPhi9ues
30 :      GOSUB 1000 'Presentation et Initialisation
40 :      GOSUB 2000 'Jeu
50 :      GOSUB 3000 'Fin de la Partie
60 :      IF FIN=0 THEN 40 'Une autre Partie est en9a9e
e
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation et Initialisation
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    GOSUB 1600 'Initialisation
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation du Jeu
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1150 :
1160 :      ATTRB 1,1 : COLOR 1 : CLS
1170 :      PRINT:PRINT" COURSE DE LEVRIERS"
1180 :
1190 :    REM Objet
1200 :
1210 :      ATTRB 0,0 : PRINT:PRINT:PRINT
1220 :      COLOR 4
1230 :      PRINT"OBJET : ";
1240 :      COLOR 6
1250 :      PRINT"Jeu de hasard."
1260 :
1270 :    REM Job
1280 :
1290 :      PRINT:PRINT:PRINT
1300 :      COLOR 5
1310 :      PRINT"JOB   : ";
1320 :      COLOR 6
1330 :      PRINT"Quatre levriers vont"
1340 :      PRINT"         s'affronter dans une course d
e"
1350 :      PRINT"         vitesse. Le but du jeu est de
"
1360 :      PRINT"         Parier sur l'ordre d'arrivee"
1370 :      PRINT"         des levriers."
```

```
1380 :      PRINT"         La course se deroulera alors"
1390 :      PRINT"         sous vos yeux."
1400 :
1410 :    REM Debut
1420 :
1430 :      PRINT:PRINT:PRINT
1440 :      COLOR 3
1450 :      PRINT"DEBUT : ";
1460 :      COLOR 6
1470 :      PRINT"Pour commencer, appuyez sur"
1480 :      PRINT"         une touche quelconque."
1490 :      A=RND : A$=INKEY$ : IF A$="" THEN 1490 'Att
ente de l'appui
1500 :
1510 :  RETURN
1520 :  REM ****************************************
1600 :  REM Initialisation des variables et constantes
du jeu
1610 :
1620 :    CLS
1625 :    DI=1000 'Somme disponible au debut du jeu
1630 :
1640 :    REM Definition des levriers
1645 :
1650 :      DEFGR$(0)=0,12,12,60,60,12,15,15
1655 :      DEFGR$(1)=0,0,0,0,0,0,224,240
1660 :      DEFGR$(2)=15,15,15,8,6,6,6,3
1665 :      DEFGR$(3)=248,252,244,116,112,96,224,0
1670 :      DEFGR$(4)=0,12,12,60,60,12,15,15
1675 :      DEFGR$(5)=0,0,0,0,0,0,224,240
1680 :      DEFGR$(6)=7,15,31,24,48,48,48,0
1685 :      DEFGR$(7)=248,244,252,56,12,12,12,12
1690 :
1700 :  RETURN
1710 :  REM ****************************************
2000 :  REM Deroulement du jeu
2010 :
2020 :    GOSUB 2100 'Initialisation du jeu
2030 :    GOSUB 2600 'Course de levriers
2040 :
2050 :  RETURN
2060 :  REM ****************************************
2100 :  REM Initialisation du jeu
2110 :
2120 :    CLS : COLOR 1
2130 :    REM Titre, commentaire et parcours
2140 :
2150 :      LOCATE 11,0 : PRINT"COURSE DE LEVRIERS"
```

```
2160 :      LOCATE 2,2 : COLOR 6
2165 :      PRINT"Arrivee                          DePart"
2170 :
2180 :    FOR I=1 TO 4
2190 :      LOCATE 2,4+I*3 : PRINT CHR$(I+48)
2200 :    NEXT I
2210 :
2220 :    FOR I=1 TO 5
2230 :      FOR J=1 TO 30
2240 :        PSET(J+3,I*3+2)"="
2250 :      NEXT J
2260 :    NEXT I
2270 :
2280 :    FOR I=0 TO 3
2290 :      FOR J=1 TO 2
2300 :        PSET(5,5+J+I*3)"I"
2305 :        PSET(32,5+J+I*3)"I"
2310 :      NEXT J
2320 :    NEXT I
2330 :
2340 :    REM Chiens au depart
2350 :
2360 :      FOR I=1 TO 4
2370 :        PSET(33,3+I*3)GR$(4) : PSET(34,3+I*3)GR$(
5)
2375 :        PSET(33,4+I*3)GR$(6) : PSET(34,4+I*3)GR$(
7)
2380 :      NEXT I
2390 :
2400 :    REM Questions au Joueur
2410 :
2420 :      LOCATE 0,19 : COLOR 3
2430 :      PRINT"Entrez l'ordre d'arrivee des levriers
:"
2440 :      LOCATE 0,21 : INPUT"Premier : ";X(1)
2450 :      LOCATE 0,21 : INPUT"Second  : ";X(2)
2460 :      LOCATE 0,21 : INPUT"Troisieme: ";X(3)
2470 :      LOCATE 0,21 : INPUT"Quatrieme: ";X(4)
2480 :      LOCATE 0,19 : PRINT SPC(119)
2490 :      LOCATE 0,19 : PRINT"Vous disPosez de "DI" f
rancs."
2500 :      PRINT : INPUT"Votre mise ";MI
2510 :      LOCATE 0,19 : PRINT SPC(119)
2520 :
2525 :    REM Debut de la course
2530 :
2540 :      COLOR 5 : PRINT"              RIEN NE VA PLUS
 ..."
2550 :      P(1)=33 : P(2)=33 : P(3)=33 : P(4)=33 'Chie
ns au depart
```

```
2555 :
2560 : RETURN
2570 : REM ******************************************
2600 : REM Course de levriers
2610 :
2615 :    COLOR 6
2620 :    D=0 'RAZ du compteur d'arrivee
2630 :    FOR I=1 TO 4
2640 :      A=INT(RND*10)
2650 :      IF A<4 OR P(I)=3 THEN 2750
2660 :      IF T(I)=0 THEN T(I)=1 ELSE T(I)=0 'Position
 repos ou galop
2670 :      B=3+I*3 : C=T(I)*4
2680 :      PSET(P(I),B)GR$(C) : PSET(P(I)+1,B)GR$(C+1)
2685 :      PSET(P(I)+2,B)" " : PSET(P(I),B+1)GR$(C+2)
2690 :      PSET(P(I)+1,B+1)GR$(C+3) : PSET(P(I)+2,B+1)
" "
2700 :      P(I)=P(I)-1
2710 :      FOR J=1 TO 4
2720 :        D=D+P(I)
2730 :      NEXT J
2740 :      IF P(I)=3 THEN AR=AR+1 : TE(I)=AR
2750 :    NEXT I
2760 :    IF P(1)<>3 OR P(2)<>3 OR P(3)<>3 OR P(4)<>3 T
HEN 2620
2770 :
2780 : RETURN
2790 : REM ******************************************
3000 : REM Fin de la partie
3010 :
3020 :    LOCATE 0,19 : PRINT SPC(160) : LOCATE 0,19
3030 :    FOR I=1 TO 3
3040 :      IF TE(I)=X(I) THEN GA=GA+1
3050 :    NEXT I
3055 :    GN=MI*(GA-1)
3060 :    IF GN<0 THEN PRINT"Vous perdez ";-GN;"francs.
"
3070 :    IF GN=0 THEN PRINT"Pas de gain."
3080 :    IF GN>0 THEN PRINT"Vous gagnez ";GN;" francs.
"
3090 :    DI=DI+GN
3100 :    IF DI<=0 THEN PRINT"Vous avez joue trop gros
!" : FIN=1 : RETURN
3110 :    PRINT"Il vous reste ";DI;" francs."
3120 :    COLOR 3 : INPUT"Une autre partie (O/N) ";R$
3130 :    IF R$="O" THEN FIN=0 ELSE FIN=1
3140 :    LOCATE 1,1,1 : CLS : COLOR 6
3150 :
3160 : RETURN
```

Pendu

PRESENTATION

Ce jeu de hasard fait également appel à la mémoire et à la réflexion du joueur. Un mot de N lettres est aléatoirement choisi dans une bibliothèque de cent mots stockée en mémoire.

La première et la dernière lettre du mot sont affichées à l'écran. Le but du jeu est de trouver les lettres manquantes en un minimum d'essais.

ORGANIGRAMME DU JEU

Le programme a la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- choix du mot à découvrir ;
- affichage des première et dernière lettres de ce mot ;
- lecture de la lettre proposée jusqu'à ce que le mot soit découvert ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et initialisation.
lignes 2000 à 2260 : jeu.
lignes 3000 à 3140 : fin du jeu.

Niveau 3 - **lignes 1100 à 1490** : présentation du jeu.
lignes 1500 à 1620 : initialisation du jeu.

IDENTIFICATION DES VARIABLES

A : pointe sur le mot à découvrir.

ESSAI : nombre de lettres proposées.

I : index de boucles.

L : longueur du mot à découvrir.

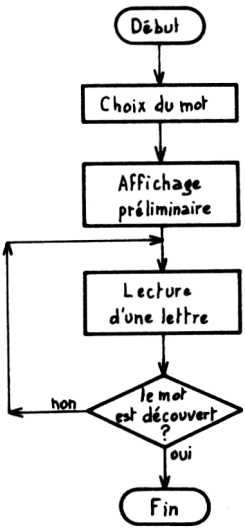
L$: lettre proposée.

R$: drapeau d'activation d'une autre partie.

TROUVE : nombre de lettres trouvées.

T$: tableau contenant la bibliothèque des mots à trouver.

ORGANIGRAMME



```
10 :    REM PENDU
20 :
30 :      GOSUB 1000 'Presentation et Initialisation
40 :      GOSUB 2000 'Jeu
50 :      GOSUB 3000 'Fin de Jeu
60 :      IF R$="O" THEN 40 'Une autre Partie est en9a9
ee
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation et Initialisation
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    GOSUB 1500 'Initialisation
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation du Jeu
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1150 :
1160 :      ATTRB 1,1 : COLOR 1 : CLS
1170 :      PRINT:PRINT"        PENDU"
1180 :
1190 :    REM Objet
1200 :
1210 :      ATTRB 0,0 : PRINT:PRINT:PRINT
1220 :      COLOR 4
1230 :      PRINT"OBJET : ";
1240 :      COLOR 6
1250 :      PRINT"Jeu de hasard et de reflexion."
1260 :
1270 :    REM Job
1280 :
1290 :      PRINT:PRINT:PRINT
1300 :      COLOR 5
1310 :      PRINT"JOB    : ";
1320 :      COLOR 6
1330 :      PRINT"MO5 va afficher la 1ere et la"
1340 :      PRINT"          derniere lettre d'un mot inco
nnu";
1350 :      PRINT"          A vous de trouver les lettres
"
1360 :      PRINT"          manquantes..."
1370 :
1380 :    REM Debut
```

```
1390 :
1400 :      PRINT:PRINT:PRINT
1410 :      COLOR 3
1420 :      PRINT"DEBUT : ";
1430 :      COLOR 6
1440 :      PRINT"Pour commencer, appuyez sur"
1450 :      PRINT"          une touche quelconque."
1460 :      A=RND:A$=INKEY$:IF A$="" THEN 1460 'Attente
 de l'appui.
1470 :
1480 : RETURN
1490 : REM ******************************************
1500 : REM Initialisation
1510 : REM Lecture de la bibliotheque de mots
1520 :
1530 :   DIM T$(100),T(20)
1540 :   FOR I=1 TO 100
1550 :     READ T$(I)
1560 :   NEXT I
1570 :
1580 :   DATA PERSONNALITE,CONTRARIETE,PETROLE,SERVIET
TE,FUGITIF
1581 :   DATA COMBINAISON,REVOLVER,ARMOIRE,STYLO,GOUVE
RNAIL
1582 :   DATA COUVERTURE,DRAP,CHATEAU,ELECTRIQUE,LIVRE
1583 :   DATA VAISSELLE,AFFICHE,GLACE,OFFICIER,PRATIQU
E
1584 :   DATA INGENIEUX,BUREAU,COLLEGE,COULOIR,PARQUET
1585 :   DATA HERBE,ARBUSTE,LAMPION,MOULAGE,SAPIN
1586 :   DATA ETAGERE,CONTROLEUR,PHYSIOLOGIE,PORTEUR,B
OUCLE
1587 :   DATA AQUARIUM,BICYCLETTE,FEUILLAGE,NAVIRE,ESC
ALIER
1588 :   DATA MLADIE,DANGER,MORTUAIRE,CARAVANE,MENAGER
IE
1589 :   DATA SALADIER,VOLCAN,VETEMENTS,FLEURISTE,VOIS
INAGE
1590 :   DATA BAVARDAGE,EMBALLAGE,CHEVET,MONTAGNE,ELEP
HANT
1591 :   DATA MUSIQUE,NATATION,COFFRET,MEDICAMENT,SILH
OUETTE
1592 :   DATA ALCHIMISTE,MACHINISTE,PHILOSOPHIE,AUTONO
ME,PALMIER
1593 :   DATA RAMPART,GUERRIER,GATERIE,DISQUETTE,REGUL
ATEUR
1594 :   DATA FAMILLE,LOINTAIN,APPAREILLER,PRESTIGIEUX
,ENCOMBRANT
1595 :   DATA CINQUANTE,PASSAGER,DESTINATION,EXTREME,L
UMIERE
```

```
1596 :    DATA ASCENSEUR,ASSIETTE,TERRESTRE,LUDIQUE,TRA
VAIL
1597 :    DATA PELUCHE,CHIRURGIEN,MOUSTACHE,BARBU,TIROI
R
1598 :    DATA PORTAIL,PLASTIQUE,TAPISSERIE,MOUILLE,CIG
ARETTE
1599 :    DATA CEINTURE,CASQUE,MOTOCYCLETTE,STARTER,AUT
ODIDACTE
1600 :
1610 : RETURN
1620 : REM ****************************************
2000 : REM Jeu
2010 :
2020 :    CLS : ATTRB 1,1 : COLOR 1
2030 :    PRINT : PRINT"        PENDU"
2040 :    A=INT(RND*100)+1 : MO$=T$(A) 'Choix du mot a
decouvrir
2050 :    L=LEN(MO$) : ATTRB 0,0 : COLOR 6
2060 :    LOCATE 0,10 : PRINT"Le mot a decouvrir compor
te"L"lettres."
2070 :    ATTRB 1,1 : COLOR 3 : LOCATE INT((40-L)/4),14
2080 :    PRINTLEFT$(MO$,1); : TROUVE=2
2090 :    FOR I=1 TO L-2
2100 :      PRINT"-";
2110 :    NEXT I
2120 :    PRINT RIGHT$(MO$,1)
2130 :
2140 :    REM Boucle Principale
2150 :
2155 :      ESSAI=ESSAI+1
2160 :      ATTRB 0,0 : COLOR 1 : LOCATE 0,18
2170 :      PRINT"Quelle lettre Proposez vous ?" : L$=I
NPUT$(1)
2180 :      ATTRB 1,1 : COLOR 3
2190 :      FOR I=2 TO L-1
2200 :        IF MID$(MO$,I,1)<>L$ THEN 2220 'Mauvaise
lettre
2205 :      IF T(I)=0 THEN T(I)=1 : TROUVE=TROUVE+1
2210 :        LOCATE INT((40-L)/4)+(I-1)*2,14 : PRINT L
$
2220 :      NEXT I
2230 :      IF TROUVE<>L THEN 2130 'Le mot n'est Pas tr
ouve
2240 :
2250 : RETURN
2260 : REM ****************************************
3000 : REM Le mot a ete trouve
3010 :
```

```
3020 :    CLS
3030 :    ATTRB 0,0 : COLOR 5
3040 :    PRINT"Mot trouve en"ESSAI"essais."
3050 :    IF ESSAI<=L+3 THEN PRINT"Bravo!" ELSE PRINT"B
of, bof ..."
3060 :    PRINT:PRINT:PRINT
3070 :    INPUT"Un autre essai";R$
3075 :    IF R$="N" THEN 3090TE 1,1,0 : ESSAI=0
3080 :    LOCATE 1,1,0 : ESSAI=0
3090 :    FOR I=1 TO 20
3100 :      T(I)=0
3110 :    NEXT I
3120 :
3130 : RETURN
3140 : REM ****************************************
```

JEUX DE RÉFLEXION

Grenouilles et Crapauds

PRESENTATION

Ce jeu, encore connu sous le nom de "saute-mouton", demande une certaine faculté de réflexion : N crapauds se trouvent séparés de N grenouilles par une case vide (où N quelconque, mais > 1).

Les crapauds sont à gauche, les grenouilles à droite. Le but du jeu est d'amener tous les crapauds à droite et toutes les grenouilles à gauche, en respectant les cinq règles suivantes :

- les crapauds ne peuvent se déplacer que vers la droite ;
- les grenouilles ne peuvent se déplacer que vers la gauche ;
- un crapaud (resp. une grenouille) peut passer sur la case située immédiatement à sa droite (resp. à sa gauche) si celle-ci est libre ;
- un crapaud peut sauter par-dessus une grenouille si la case située à sa droite est libre ;
- une grenouille peut sauter par-dessus un crapaud si la case située à sa gauche est libre.

ORGANIGRAMME DU JEU

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- saisie du déplacement ;
- test sur le déplacement ;
- déplacement effectué ou refusé ;
- test si la partie est finie ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 80** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et initialisation.
lignes 2000 à 2060 : jeu.
lignes 3000 à 3060 : fin du jeu.

Niveau 3 - **lignes 1100 à 1620** : présentation du jeu.
lignes 1700 à 1920 : initialisation du jeu.
lignes 2100 à 2396 : déplacement.
lignes 2400 à 2480 : affichage du jeu.
lignes 2900 à 2990 : test fin de partie.

IDENTIFICATION DES VARIABLES

A$: attente d'un caractère pour passer à la suite.

COUPS : nombre de coups joués.

DE $: code de déplacement.

I : index de boucles.

JEU $: tableau de jeu.

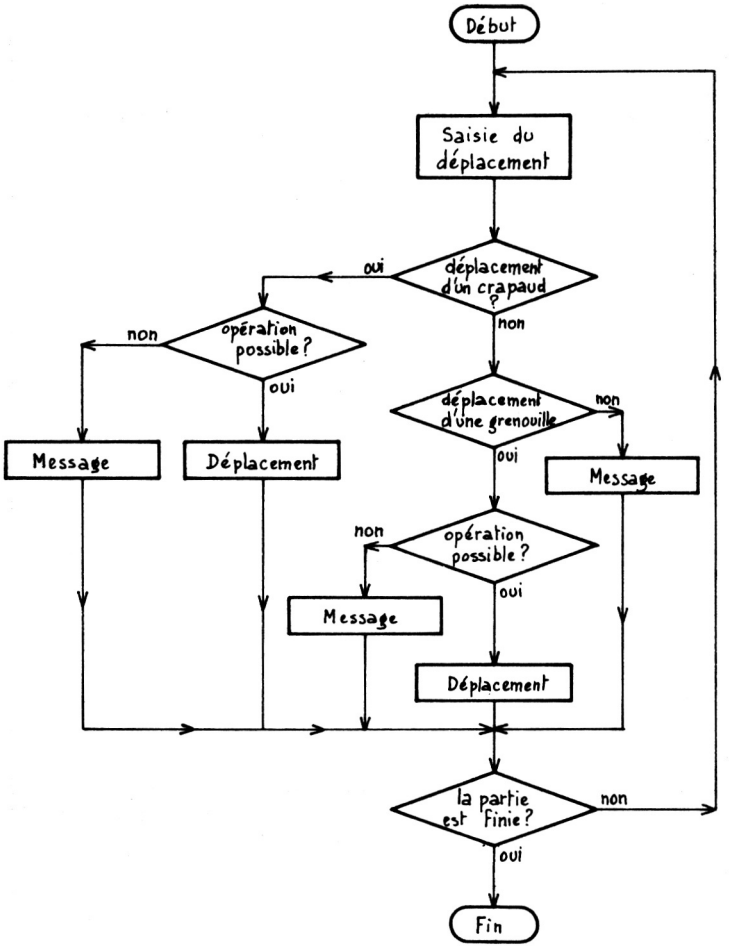
NC : nombre de crapauds.

NG : nombre de grenouilles.

NER : nombre d'erreurs.

PTR : indicateur de déplacement.

ORGANIGRAMME



```
10 :   REM GRENOUILLES ET CRAPAUDS
20 :
30 :     GOSUB 1000 'Presentation et Initialisation
40 :     GOSUB 2000 'Jeu
50 :     GOSUB 3000 'Fin de jeu
60 :
70 :   END
80 :   REM ****************************************
1000 : REM Presentation et Initialisation
1010 :
1020 :   GOSUB 1100 'Presentation
1030 :   GOSUB 1700 'Initialisation
1040 :
1050 : RETURN
1060 : REM ****************************************
1100 : REM Presentation du jeu
1110 :
1120 :   SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :   REM Titre
1150 :
1160 :     ATTRB 0,1 : COLOR 1 : CLS
1170 :     PRINT:PRINT"       GRENOUILLES ET CRAPAUDS"
1180 :
1190 :   REM Objet
1200 :
1210 :     ATTRB 0,0
1220 :     COLOR 4
1230 :     PRINT"OBJET : ";
1240 :     COLOR 6
1250 :     PRINT"Jeu de reflexion."
1260 :
1270 :   REM Job
1280 :
1290 :     PRINT
1300 :     COLOR 5
1310 :     PRINT"JOB    : ";
1320 :     COLOR 6
1330 :     PRINT"Les crapauds sont definis par"
1340 :     PRINT"         la lettre C, les grenouilles"
1350 :     PRINT"         par la lettre G."
1360 :     PRINT"         Le but du jeu est d'amener en
"
1370 :     PRINT"         un minimum de coups les"
1390 :     PRINT"         crapauds a droite et les"
1400 :     PRINT"         grenouilles a gauche."
1410 :     PRINT"         Restrictions: Un crapaud"
1420 :     PRINT"         peut se deplacer d'une case"
1430 :     PRINT"         vers la droite si elle est"
```

```
1440 :      PRINT"          libre(DC). Il peut egalement"
1450 :      PRINT"          sauter par dessus une"
1460 :      PRINT"          grenouille si la case situee"
1470 :      PRINT"          a sa droite est libre(DC)."
1480 :      PRINT"          De meme pour une grenouille,"
1490 :      PRINT"          mais vers la gauche(DG et SG)
"
1500 :
1510 :    REM Debut
1520 :
1530 :      PRINT
1540 :      COLOR 3
1550 :      PRINT"DEBUT : ";
1560 :      COLOR 6
1570 :      PRINT"Pour commencer, appuyez sur"
1580 :      PRINT"          une touche quelconque."
1590 :      A$=INPUT$(1) 'Attente de l'appui
1600 :
1610 :  RETURN
1620 :  REM ****************************************
1700 :  REM Initialisation
1710 :
1720 :    CLS
1730 :    LOCATE 0,10
1740 :    INPUT"Nombre de crapauds ";NC
1750 :    DIM JEU$(2*NC+1)
1755 :
1760 :    REM Constitution du tableau de jeu
1770 :
1780 :      FOR I=1 TO NC
1790 :        JEU$(I)="C"
1800 :      NEXT I
1810 :      JEU$(NC+1)=" "
1820 :      FOR I=NC+2 TO 2*NC+1
1830 :        JEU$(I)="G"
1840 :      NEXT I
1845 :
1850 :    REM Affichage
1855 :
1860 :      CLS : LOCATE 0,10
1870 :      FOR I=1 TO NC*2+1
1875 :        IF JEU$(I)="C" THEN COLOR 3 ELSE COLOR 1
1880 :        PRINT JEU$(I);
1890 :      NEXT I
1895 :      COLOR 6
1900 :
1910 :  RETURN
1920 :  REM ****************************************
2000 :  REM Traitement
```

```
2010 :
2020 :    GOSUB 2100 'Deplacements
2030 :    GOSUB 2900 'Fin de Partie
2040 :    IF NOFIN=1 THEN 2000 'Boucle de Partie
2045 :
2050 : RETURN
2060 : REM ****************************************
2100 : REM Deplacements
2110 :
2120 :    LOCATE 0,18 : PRINT SPC(120)
2130 :    LOCATE 0,18 : INPUT"Deplacement";DE$ : PTR=-1
 : COUPS=COUPS+1
2140 :    IF DE$<>"DC" THEN 2190
2141 :
2142 :    REM Deplacement Crapaud
2143 :
2145 :       FOR I=1 TO NC*2
2150 :         IF JEU$(I)="C" AND JEU$(I+1)=" " THEN PTR
=I
2160 :       NEXT I
2170 :       IF PTR=-1 THEN GOSUB 2370 : RETURN 'Deplace
ment inacceptE
2180 :       JEU$(PTR)=" " : JEU$(PTR+1)="C" : GOSUB 240
0 : RETURN 'Depl. accepte
2190 :       IF DE$<>"SC" THEN 2240
2191 :
2192 :    REM Saut Crapaud
2193 :
2195 :       FOR I=1 TO NC*2-1
2200 :         IF JEU$(I)="C" AND JEU$(I+1)="G" AND JEU$
(I+2)=" " THEN PTR=I
2210 :       NEXT I
2220 :       IF PTR=-1 THEN GOSUB 2370 : RETURN 'Deplace
ment inacceptE
2230 :       JEU$(PTR)=" " : JEU$(PTR+2)="C" : GOSUB 240
0 : RETURN 'Depl. accepte
2240 :       IF DE$<>"DG" THEN 2310
2250 :
2260 :    REM Deplacement Grenouille
2261 :
2265 :       FOR I=NC*2+1 TO 2 STEP -1
2270 :         IF JEU$(I)="G" AND JEU$(I-1)=" " THEN PTR
=I
2280 :       NEXT I
2290 :       IF PTR=-1 THEN GOSUB 2370 : RETURN 'Deplace
ment inacceptE
2300 :       JEU$(PTR)=" " : JEU$(PTR-1)="G" : GOSUB 240
0 : RETURN 'Depl. accepte
2310 :       IF DE$<>"SG" THEN 2370
```

```
2311 :
2312 :    REM Saut Grenouille
2313 :
2315 :       FOR I=NC*2+1 TO 3 STEP -1
2320 :         IF JEU$(I)="G" AND JEU$(I-1)="C" AND JEU$
(I-2)=" " THEN PTR=I
2330 :       NEXT I
2340 :       IF PTR=-1 THEN GOSUB 2370 : RETURN 'Deplace
ment inacceпte
2350 :       JEU$(PTR)=" " : JEU$(PTR-2)="G" : GOSUB 240
0 : RETURN 'Depl. acceпte
2360 :
2370 :    REM Deplacement inacceпte
2380 :
2385 :       COLOR 1 : LOCATE 0,20 : PRINT"Deplacement i
naccepte" : COLOR 6
2390 :       NER=NER+1 : PLAY"SIPSI"
2391 :
2395 :  RETURN
2396 :  REM ****************************************
2400 :  REM Le deplacement est acceпte
2410 :
2420 :    LOCATE 0,10
2430 :    FOR I=1 TO 2*NC+1
2435 :      IF JEU$(I)="C" THEN COLOR 3 ELSE COLOR 1
2440 :      PRINT JEU$(I);
2450 :    NEXT I
2455 :    COLOR 6
2460 :
2470 :  RETURN
2480 :  REM ****************************************
2900 :  REM Test si fin de Partie
2910 :
2920 :    NG=0 'RAZ variable de fin de jeu
2930 :    FOR I=1 TO NC
2940 :      IF JEU$(I)="G" THEN NG=NG+1
2950 :    NEXT I
2960 :    IF NG=NC AND JEU$(NC+1)=" " THEN NOFIN=0 ELSE
 NOFIN=1
2970 :
2980 :  RETURN
2990 :  REM ****************************************
3000 :  REM Fin de la Partie
3010 :
3020 :    CLS
3030 :    PRINT"Partie terminee en"COUPS"coups,"
3040 :    PRINT"avec"NER"erreur(s)."
3050 :
3060 :  RETURN
```

Tours de Hanoï

PRESENTATION

Ce jeu de réflexion a suscité et suscite encore beaucoup d'écrits. Il comprend trois tours. La tour de gauche comporte plusieurs disques empilés, de diamètres différents, le plus petit en haut et le plus grand en bas.

Le but du jeu est de restituer l'empilage de la tour de gauche sur la tour de droite en respectant la règle suivante : un disque ne peut se poser que sur une tour vide ou un disque plus grand.

ORGANIGRAMME DU JEU

Le programme a la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en cinq actions élémentaires :

- affichage des tours au départ ;
- saisie des tours de départ et d'arrivée ;
- test de validité du déplacement ;
- déplacement ;
- test fin de jeu ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1420** : présentation du jeu.
lignes 2000 à 2060 : initialisation du jeu et jeu.
lignes 3000 à 3060 : fin du jeu.

Niveau 3 - **lignes 2100 à 2250** : initialisation du jeu.
lignes 2300 à 2800 : jeu.

IDENTIFICATION DES VARIABLES

ARRIVEE : drapeau de test tour d'arrivée.

BP : nombre de disques bien placés sur la tour de droite.

DEPART : drapeau de tour de départ.

I,J : index de boucles.

NC : nombre de coups joués.

ND : nombre de disques.

OK : drapeau de test validité du déplacement.

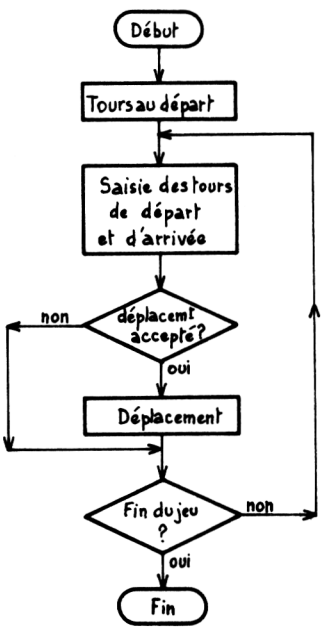
R$: indicateur de nouvelle partie.

T : tableau représentant l'empilement sur chaque tour.

TA : tour d'arrivée.

TD : tour de départ.

ORGANIGRAMME



```
10 :   REM TOURS DE HANOI
20 :
30 :     GOSUB 1000 'Presentation
40 :     GOSUB 2000 'Jeu
50 :     GOSUB 3000 'Fin du jeu
60 :     IF R$="O" THEN 40 'Nouvelle Partie
70 :
80 : END
90 : REM ****************************************
1000 : REM Presentation
1010 :
1020 :   SCREEN 6,0,0 : LOCATE 1,1,0
1030 :
1040 :   REM Titre
1050 :
1060 :     CLS : COLOR 1 : ATTRB 1,1
1070 :     PRINT : PRINT"   TOURS DE HANOI"
1080 :
1090 :   REM Objet
1100 :
1110 :     ATTRB 0,0 : PRINT:PRINT:PRINT
1120 :     COLOR 4
1130 :     PRINT"OBJET : ";
1140 :     COLOR 6
1150 :     PRINT"Jeu de reflexion."
1160 :
1170 :   REM Job
1180 :
1190 :     PRINT:PRINT:PRINT
1200 :     COLOR 5
1210 :     PRINT"JOB    : ";
1220 :     COLOR 6
1230 :     PRINT"N disques vont apparaitre"
1240 :     PRINT"         sur la tour de gauche (ou N e
st"
1250 :     PRINT"         choisi par le joueur). Vous "
1260 :     PRINT"         devrez reconstituer l'empilem
ent";
1270 :     PRINT"         de la tour de gauche sur la t
our";
1280 :     PRINT"         de droite, sachant qu'un disq
ue"
1290 :     PRINT"         ne peut se poser que sur un"
1300 :     PRINT"         autre de taille superieure."
1310 :
1320 :   REM Debut
1330 :
1335 :     PRINT:PRINT:PRINT
```

```
1340 :      COLOR 3
1350 :      PRINT"DEBUT : ";
1360 :      COLOR 6
1370 :      PRINT"Pour commencer, appuyez sur"
1380 :      PRINT"          une touche quelconque."
1390 :      A$=INPUT$(1) 'Attente de l'appui
1391 :      DIM T(9,4) 'Tableau de jeu
1400 :
1410 : RETURN
1420 : REM ******************************************
2000 : REM Jeu
2010 :
2020 :   GOSUB 2100 'Initialisation
2030 :   GOSUB 2300 'Partie
2040 :
2050 : RETURN
2060 : REM ******************************************
2100 : REM Initialisation du jeu
2110 :
2120 :   CLS
2130 :   ATTRB 1,1 : COLOR 1
2140 :   PRINT : PRINT"   TOURS DE HANOI"
2150 :   ATTRB 0,0 : LOCATE 0,10 : COLOR 6
2160 :   INPUT"Nombre de disques(entre 3 et 7)";ND
2170 :   IF ND>7 OR ND<3 THEN PLAY"SIPSI" : GOTO 2150
'Entree non acceptee
2180 :
2181 :   REM Initialisation du tableau de jeu
2182 :
2190 :   FOR J=0 TO 3
2191 :     FOR I=0 TO 8
2192 :       T(I,J)=0
2193 :     NEXT I
2194 :   NEXT J
2195 :
2200 :   FOR I=1 TO ND
2210 :     T(I,1)=I
2220 :   NEXT I
2230 :
2240 : RETURN
2250 : REM ******************************************
2300 : REM Deroulement d'une Partie
2310 :
2320 :   CLS
2330 :   ATTRB 1,1 : COLOR 1
2340 :   PRINT : PRINT"   TOURS DE HANOI"
2350 :
2360 :   REM Trace des disques
```

```
2370 :
2375 :      COLOR 6
2380 :      FOR J=1 TO 3
2390 :        FOR I=1 TO ND
2400 :          IF T(I,J)=0 THEN 2460
2410 :          A=(J-1)*95+50-5*(8+T(I,J)-ND)
2420 :          B=150-12*(ND-I+1)
2430 :          C=(J-1)*95+60+5*(8+T(I,J)-ND)
2440 :          D=B+12
2450 :          BOX(A,B)-(C,D)
2460 :        NEXT I
2470 :      NEXT J
2480 :
2490 :    REM Saisie du deplacement
2500 :
2510 :      COLOR 3 : ATTRB 0,0
2520 :      LOCATE 0,20 : INPUT"Tour de depart ";TD
2530 :      LOCATE 0,22 : INPUT"Tour d'arrivee ";TA
2540 :
2550 :    REM Test de la validite de la demande
2560 :
2565 :      IF TA>3 OR TD>3 OR TA<1 OR TD<1 THEN OK=1 :
 GOTO 2640
2570 :      I=ND
2580 :      IF T(I,TA)<>0 THEN I=I-1 : GOTO 2580
2590 :      J=ND
2600 :      IF T(J,TD)<>0 THEN J=J-1 : GOTO 2600
2610 :      J=J+1 : OK=1
2620 :      IF I=ND AND J<=ND THEN OK=-1 : GOTO 2640
2625 :      IF J>ND THEN OK=1 : GOTO 2640
2630 :      IF T(I+1,TA)>T(J,TD) THEN OK=-1
2640 :      IF OK=1 THEN COLOR 1 : PRINT"Deplacement no
n accepte." : PLAY"SIPSI" : COLOR 3 : GOTO 2520
2650 :
2660 :    REM Memorisation du deplacement
2670 :
2680 :      T(I,TA)=T(J,TD) : T(J,TD)=0
2690 :
2700 :    REM Test si la partie est terminee
2710 :
2720 :      BP=0 'Bien positionne = 0
2730 :      FOR I=1 TO ND
2740 :        IF T(I,3)=I THEN BP=BP+1
2750 :      NEXT I
2751 :
2752 :      NC=NC+1 'Nombre de coups joues
2760 :
2770 :      IF BP<>ND THEN 2300 'La partie n'est pas fi
nie
```

```
2780 :
2790 : RETURN
2800 : REM ******************************************
3000 : REM Fin du jeu
3010 :
3020 :   CLS : LOCATE 0,10
3030 :   PRINT"Tours de HANOI deplacees en"NC"coups."
3040 :   PRINT : INPUT"Un autre essai (O/N) ";R$
3050 :
3060 : RETURN
```

Taquin

PRESENTATION

Ce jeu, inventé aux Etats-Unis le siècle dernier, est mondialement connu. Il se compose d'un carré comportant quinze plaquettes numérotées de un à quinze et une plaquette vide.

Le but du jeu est de bouger les quinze plaquettes en se servant de la case vide pour les mettre dans l'ordre croissant. Au départ, les plaquettes se trouvent dans un ordre quelconque.

ORGANIGRAMME DU JEU

Le programme a la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- saisie de la case à déplacer ;
- test de validité de la demande ;
- test fin de jeu ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et initialisation.
lignes 2000 à 2580 : jeu.
lignes 3000 à 3070 : fin du jeu.

Niveau 3 - **lignes 1100 à 1490** : présentation du jeu.
lignes 1500 à 1720 : initialisation du jeu.

IDENTIFICATION DES VARIABLES

A : tirage aléatoire du taquin initial.

BIS : indicateur de mauvais tirage.

BP : nombre de cases bien placées.

CD : case à déplacer.

I,J : index de boucles.

ID,JD : position de la case libre sur le taquin.

IJ,JJ : position de la case à déplacer sur le taquin.

NC : nombre de coups joués.

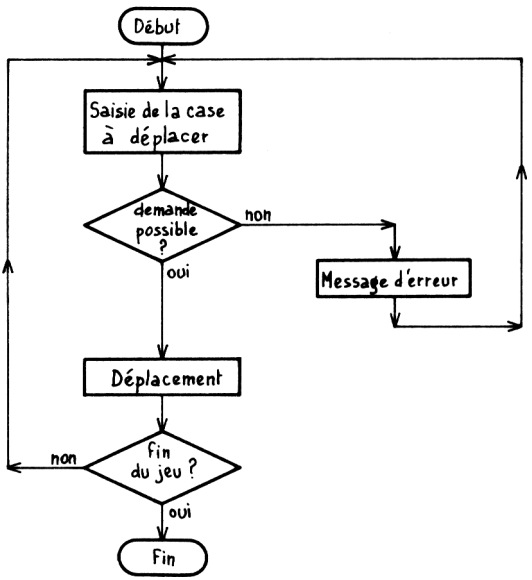
OK : indicateur de déplacement possible.

R$: drapeau d'activation d'une nouvelle partie.

T : tableau contenant le taquin initial.

TJ : tableau de jeu.

ORGANIGRAMME



```
10 :    REM TAQUIN
20 :
25 :      CLEAR 'Toutes les declarations sont annulees
30 :      GOSUB 1000 'Presentation et Initialisation
40 :      GOSUB 2000 'Jeu
50 :      GOSUB 3000 'Fin de Jeu
60 :      IF R$="O" THEN 10 'Une autre Partie est en9a9
ee
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation et Initialisation
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    GOSUB 1500 'Initialisation
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation du Jeu
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1150 :
1160 :      ATTRB 1,1 : COLOR 1 : CLS
1170 :      PRINT:PRINT"       TAQUIN"
1180 :
1190 :    REM Objet
1200 :
1210 :      ATTRB 0,0 : PRINT:PRINT:PRINT
1220 :      COLOR 4
1230 :      PRINT"OBJET : ";
1240 :      COLOR 6
1250 :      PRINT"Jeu de reflexion."
1260 :
1270 :    REM Job
1280 :
1290 :      PRINT:PRINT:PRINT
1300 :      COLOR 5
1310 :      PRINT"JOB    : ";
1320 :      COLOR 6
1330 :      PRINT"Reconstituez le Taquin en un"
1340 :      PRINT"         minimum de coups en deplacant
"
1350 :      PRINT"         les 15 plaquettes par rapport
 a"
1360 :      PRINT"         la plaquette vide."
1370 :
```

→

```
1380 :    REM Debut
1390 :
1400 :      PRINT:PRINT:PRINT
1410 :      COLOR 3
1420 :      PRINT"DEBUT : ";
1430 :      COLOR 6
1440 :      PRINT"Pour commencer, appuyez sur "
1450 :      PRINT"         une touche quelconque."
1460 :      A=RND : A$=INKEY$ : IF A$="" THEN 1460 'Att
ente de l'appui
1470 :
1480 : RETURN
1490 : REM ****************************************
1500 : REM Initialisation du jeu et de l'affichage
1510 :
1520 :    REM Initialisation du jeu
1530 :
1540 :      DIM T(16),TJ(6,6)
1550 :      FOR I=1 TO 16 'Tirage des 15 nombres du taq
uin
1560 :        A=INT(RND*16) : BIS=1
1570 :        IF I=1 THEN 1620'Pas de test de repetitio
n sur le 1er nombre
1580 :        FOR J=1 TO I-1
1590 :          IF T(J)=A THEN BIS=-1
1600 :        NEXT J
1610 :        IF BIS=-1 THEN 1560 'Le nombre a deja ete
 tire
1620 :        T(I)=A
1630 :      NEXT I
1640 :
1650 :      FOR I=1 TO 4
1660 :        FOR J=1 TO 4
1670 :          TJ(I,J)=T((I-1)*4+J)
1680 :        NEXT J
1690 :      NEXT I
1700 :
1710 : RETURN
1720 : REM ****************************************
2000 : REM Jeu
2005 :
2010 :    REM Initialisation de l'affichage
2020 :
2030 :      ATTRB 1,1 : COLOR 1 : CLS
2040 :      PRINT : PRINT"         TAQUIN"
2050 :
2055 :      COLOR 3 : ATTRB 0,0
2060 :      FOR I=0 TO 3
```

```
2070 :        FOR J=0 TO 3
2080 :          LOCATE 13+J*3,16+I*2
2090 :          A=TJ(I+1,J+1) : IF A<> 0 THEN PRINT A
2100 :        NEXT J
2110 :      NEXT I
2120 :
2125 :      COLOR 6
2130 :      FOR I=0 TO 4
2140 :        LINE(108,124+I*16)-(204,124+I*16)
2150 :        LINE(108+I*24,124)-(108+I*24,188)
2160 :      NEXT I
2170 :
2180 :    REM Saisie de la case a deplacer
2190 :
2200 :      COLOR 5
2210 :      LOCATE 10,10
2220 :      INPUT"Case a deplacer ";CD
2230 :
2240 :    REM Test validite de la demande
2250 :
2260 :      OK=1
2270 :      FOR I=1 TO 4
2280 :        FOR J=1 TO 4
2290 :          IF TJ(I,J)=CD THEN IJ=I : JJ=J
2300 :        NEXT J
2310 :      NEXT I
2320 :
2330 :      IF IJ<>4 AND TJ(IJ+1,JJ)=0 THEN OK=-1 : ID=
IJ+1 : JD=JJ
2340 :      IF IJ<>1 AND TJ(IJ-1,JJ)=0 THEN OK=-1 : ID=
IJ-1 : JD=JJ
2350 :      IF JJ<>4 AND TJ(IJ,JJ+1)=0 THEN OK=-1 : ID=
IJ : JD=JJ+1
2360 :      IF JJ<>1 AND TJ(IJ,JJ-1)=0 THEN OK=-1 : ID=
IJ : JD=JJ-1
2370 :
2380 :      IF OK=-1 THEN 2420 'La demande est acceptee
2390 :      COLOR 1 : PLAY"SIPSI"
2400 :      PRINT : PRINT"          Deplacement impossib
le !"
2410 :      GOTO 2180
2415 :
2420 :    REM Deplacement accepte
2430 :
2440 :      TJ(ID,JD)=TJ(IJ,JJ) : TJ(IJ,JJ)=0
2450 :
2460 :    REM Test fin de jeu
2470 :
```

```
2480 :      BP=0
2490 :      FOR I=1 TO 4
2500 :        FOR J=1 TO 4
2510 :          IF TJ(I,J)=(I-1)*4+J THEN BP=BP+1
2520 :        NEXT J
2530 :      NEXT I
2540 :      NC=NC+1 'Nombre de coups joues
2550 :      IF BP<>15 THEN 2000 'Le jeu n'est pas termi
ne
2560 :
2570 : RETURN
2580 : REM ****************************************
3000 : REM Le jeu est fini
3010 :
3020 :    CLS : COLOR 6
3030 :    PRINT"Partie terminee en"NC"coups."
3040 :    PRINT:PRINT
3050 :    INPUT"Une autre partie (O/N)";R$
3060 :
3070 : RETURN
```

Awari

PRESENTATION

AWARI est un jeu de réflexion à deux joueurs. Vous jouez contre l'ordinateur. Le jeu se déroule sur un parcours rectangulaire. Vous occupez la partie basse et MO5 la partie haute. Ce parcours est divisé en cases. Chaque case contient un nombre. Au début de la partie, ce nombre est trois pour chaque case. A tour de rôle, vous désignez (resp. MO5 désigne) une case de votre (resp. de son) camp. Cette case est mise à Ø. Si son contenu était n, les n cases situées en aval (dans une rotation de sens trigo) sont incrémentées de 1. Les cases latérales constituent les totaux joueur et MO5. Si l'incrémentation finit sur une telle case, le joueur concerné rejoue.

La partie se termine quand le joueur ou MO5 a annulé toutes les cases de son camp.

La partie est gagnée pour le joueur si :

- toutes les cases de la partie basse sont à Ø ;
- la case total de droite est supérieure à la case total de gauche ;

ORGANIGRAMME DU JEU

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- jeu joueur ;
- test nouveau jeu ;
- test fin de partie ;
- jeu ordinateur ;

- test nouveau jeu ;
- test fin de partie ;

d'où l'organigramme de la page suivante.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1470** : présentation du jeu.
lignes 2000 à 2060 : initialisation du jeu et jeu.
lignes 6000 à 6160 : fin du jeu.

Niveau 3 - **lignes 2100 à 2440** : initialisation du jeu.
lignes 2500 à 2600 : jeu.
lignes 3000 à 3170 : joueur.
lignes 4000 à 4280 : ordinateur.
lignes 3200 à 3260 : déplacement vers la droite.
lignes 3300 à 3360 : déplacement vers la gauche.
lignes 3400 à 3570 : flèche positionnée.
lignes 5000 à 5310 : calcul et affichage du jeu MO5.

IDENTIFICATION DES VARIABLES

A : variable de calcul.

A$: attente d'un caractère pour passer à la suite.

B$: acquisition clavier.

C, D : choix du meilleur coup.

CASE : numéro de la case choisie.

E : variable d'affichage.

GA : nom du vainqueur.

I, J : index de boucles.

JEU : indicateur de jeu (MO5 ou joueur).

PX : position de la flèche.

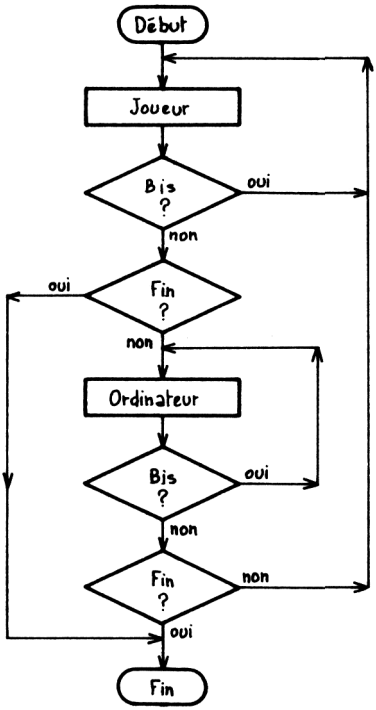
SO : total d'un des camps.

T : tableau de jeu.

Remarque

Pour augmenter la complexité du jeu, modifier les lignes 2290 à 2380 : remplacer le "3" par un nombre supérieur.

ORGANIGRAMME




```
10 :    REM AWARI
20 :
25 :      CLEAR,,2 'Declaration de 2 motifs graPhiques
30 :      GOSUB 1000 'Presentation
40 :      GOSUB 2000 'Initialisation du jeu et jeu
50 :      GOSUB 6000 'Fin de la Partie
60 :      IF R$="O" THEN 10 'Nouvelle Partie
70 :
80 : END
90 : REM ****************************************
1000 : REM Presentation
1010 :
```

```
1020 :    SCREEN 6,0,0 : LOCATE 1,1,0
1030 :
1040 :    REM Titre
1050 :
1060 :      CLS : COLOR 1 : ATTRB 1,1
1070 :      PRINT : PRINT"        AWARI"
1080 :
1090 :    REM Objet
1100 :
1110 :      ATTRB 0,0 : PRINT
1120 :      COLOR 4
1130 :      PRINT"OBJET : ";
1140 :      COLOR 6
1150 :      PRINT"Jeu de reflexion."
1160 :
1170 :    REM Job
1180 :
1190 :      PRINT
1200 :      COLOR 5
1210 :      PRINT"JOB    : ";
1220 :      COLOR 6
1230 :      PRINT"Le but du jeu est d'annuler"
1240 :      PRINT"          le nombre de points dans son"
1250 :      PRINT"          camP(Vous en bas, MO5 en haut
),"
1260 :      PRINT"          en tenant compte des regles :
"
1270 :      PRINT"            1. Le jeu d'1 case contenan
t"
1280 :      PRINT"          n Pions Provoque n additions"
1290 :      PRINT"          d'1 Pion sur les n cases"
1300 :      PRINT"          situees en aval(Sens trigo)."
1310 :      PRINT"            2. Si la derniere addition"
1320 :      PRINT"          tombe sur une case 'total',"
1330 :      PRINT"          le Joueur(vous ou MO5) rejoue
"
1340 :      PRINT"          immediatement."
1350 :
1360 :    REM Debut
1370 :
1380 :      PRINT
1390 :      COLOR 3
1400 :      PRINT"DEBUT : ";
1410 :      COLOR 6
1420 :      PRINT"Pour commencer, appuyez sur"
1430 :      PRINT"          une touche quelconque."
1440 :      A$=INPUT$(1) 'Attente de l'appui
1450 :
```

```
1460 : RETURN
1470 : REM ****************************************
2000 : REM Initialisation du jeu et jeu
2010 :
2020 :   GOSUB 2100 'Initialisation du jeu
2030 :   GOSUB 2500 'Jeu
2040 :
2050 : RETURN
2060 : REM ****************************************
2100 : REM Initialisation du jeu
2110 :
2120 :   REM Definition des caracteres graphiques
2130 :
2140 :     DEFGR$(0)=0,0,34,54,28,8,0,0
2150 :     DEFGR$(1)=0,0,8,28,54,34,0,0
2160 :
2170 :   REM Affichage du jeu
2180 :
2190 :     CLS : SCREEN 1,6,6
2200 :     ATTRB 1,1
2210 :     PRINT : PRINT"        AWARI"
2220 :
2230 :     ATTRB 0,0 : COLOR 4
2240 :     LOCATE 0,12
2250 :
2260 :     FOR I=0 TO 1
2270 :       FOR J=0 TO 5
2290 :         PSET(9+J*4,6+I*6)"3"
2300 :       NEXT J
2310 :     NEXT I
2330 :     PSET(5,9)"0"   : PSET(33,9)"0"
2340 :
2350 :     DIM T(14) 'Tableau de jeu
2360 :
2370 :     FOR I=1 TO 14
2380 :       T(I)=3
2390 :     NEXT I
2400 :
2410 :     T(1)=0 : T(8)=0 'Totaux MO5 et joueur
2420 :
2430 : RETURN
2440 : REM ****************************************
2500 : REM Jeu
2510 :
2520 :   GOSUB 3000 'Jeu joueur
2530 :   IF FIN=1 THEN RETURN 'La partie est finie
2540 :   IF BIS=1 THEN 2520 'Le joueur rejoue
2550 :   GOSUB 4000 'Jeu MO5
2560 :   IF BIS=1 THEN 2550 'MO5 rejoue
```

```
2570 :    IF FIN=0 THEN 2500 'La Partie n'est Pas finie
2580 :
2590 : RETURN
2600 : REM ******************************************
3000 : REM Jeu du Joueur
3010 :
3020 :    PSET(9,13)GR$(1) : PX=0 : BIS=0 : SO=0
3030 :    COLOR 7
3040 :    LOCATE 10,4 : PRINT"A vous de Jouer."
3050 :    LOCATE 0,16 : PRINT"Servez vous des touches f
leches droite"
3060 :    PRINT"et 9auche Pour Positionner la case"
3070 :    PRINT"desiree. FraPez "T" 9uand vous"
3080 :    PRINT"serez Positionne." : COLOR 4
3090 :
3100 :    B$=INKEY$ : IF B$<>"" THEN B=ASC(B$)
3115 :    IF B=9 THEN GOSUB 3200 'DePlacement vers la d
roite
3120 :    IF B=8 THEN GOSUB 3300 'DePlacement vers la 9
auche
3125 :    IF B$="T" THEN GOSUB 3400 'La case est select
ionnee
3130 :    IF B$="T" THEN RETURN 'Le Joueur a fini son t
our
3135 :
3140 :    GOTO 3100 'Boucle de Jeu du Joueur
3150 :
3160 : RETURN
3170 : REM ******************************************
3200 : REM DePlacement vers la droite
3210 :
3220 :    IF PX=5 THEN RETURN
3230 :    PSET(9+PX*4,13)" " : PX=PX+1 : PSET(9+PX*4,13
)GR$(1) : B=0
3240 :
3250 : RETURN
3260 : REM ******************************************
3300 : REM DePlacement vers la 9auche
3310 :
3320 :    IF PX=0 THEN RETURN
3330 :    PSET(9+PX*4,13)" " : PX=PX-1 : PSET(9+PX*4,13
)GR$(1) : B=0
3340 :
3350 : RETURN
3360 : REM ******************************************
3400 : REM La fleche est Positionnee
3410 :
3420 :    REM Effacement des informations ecrites sur l
'ecran
```

```
3430 :
3440 :      LOCATE 0,4  : PRINT SPC(40)
3450 :      LOCATE 0,16 : PRINT SPC(160)
3455 :      LOCATE 9,13 : PRINT SPC(30)
3460 :
3470 :    REM Calcul et affichage du coup joue
3480 :
3490 :      JEU=0 : GOSUB 5000
3500 :
3510 :      FOR I=2 TO 7
3520 :        SO=SO+T(I)
3530 :      NEXT I
3540 :      IF SO=0 THEN FIN=1 : GA=1
3550 :
3560 : RETURN
3570 : REM ****************************************
4000 : REM Jeu de l'ordinateur
4010 :
4020 :    MAX=-100 : BIS=0 : SO=0
4030 :    REM apres calcul, MAX contiendra le maximum
4040 :    REM de points que peut faire MO5 en 1 coup
4050 :
4060 :      FOR I=9 TO 14
4070 :        C=0 : D=0
4080 :        IF T(I)=0 THEN 4140
4085 :        FOR J=1 TO T(I)
4090 :          D=I+J
4100 :          IF D<15 AND D>8 THEN C=C-1
4110 :          IF D>1 AND D<8 THEN C=C+1
4120 :        NEXT J
4130 :        IF C>MAX THEN MAX=C : CASE=I
4140 :      NEXT I
4150 :      PX=CASE-2
4160 :
4170 :      LOCATE 0,4 : PRINT"Je joue en"
4180 :
4190 :      JEU=1 : GOSUB 5000
4200 :      LOCATE 0,4 : PRINT SPC(80)
4210 :
4220 :      FOR I=9 TO 14
4230 :        SO=SO+T(I)
4240 :      NEXT I
4250 :      IF SO=0 THEN FIN=1 : GA=0
4260 :
4270 : RETURN
4280 : REM ****************************************
5000 : REM Calcul et affichage du coup joue
5010 :
5020 :    REM Calcul
```

```
5030 :
5040 :      A=PX+2 : IF T(A)=0 THEN RETURN
5050 :      FOR I=1 TO T(A)
5060 :        IF A+I<=14 THEN T(A+I)=T(A+I)+1
5070 :        IF A+I>14  THEN T(A+I-14)=T(A+I-14)+1
5080 :      NEXT I
5090 :      IF A+I=9 OR A+I=16 THEN BIS=1
5100 :      T(A)=0
5110 :      IF JEU=1 THEN PSET(9+(14-A)*4,5)GR$(0) : FO
R J=1 TO 600 : NEXT J
5120 :
5130 :    REM Affichage
5140 :
5150 :      FOR I=0 TO 5
5160 :        LOCATE I*4+8,6
5170 :        PRINT T(14-I)
5180 :      NEXT I
5190 :
5200 :      FOR I=0 TO 5
5210 :        LOCATE I*4+8,12
5220 :        PRINT T(2+I)
5230 :      NEXT I
5240 :
5250 :      LOCATE 4,9  : PRINT T(1)
5260 :      LOCATE 32,9 : PRINT T(8)
5270 :
5280 :    IF JEU=1 THEN PSET(9+(14-A)*4,10)" "
5290 :
5300 :  RETURN
5310 :  REM ****************************************
6000 :  REM Fin de la Partie
6010 :
6020 :    SCREEN 6,0,0
6030 :    CLS
6040 :    PRINT"Partie terminee."
6050 :    PRINT:PRINT:PRINT
6060 :    IF GA=1 THEN PRINT"Bravo"; ELSE PRINT"Bof";
6070 :    PRINT" pour le jeu."
6080 :    PRINT:PRINT
6090 :    IF T(1)<T(8)THEN PRINT"Bravo"; ELSE PRINT"Bof
";
6100 :    PRINT" pour les points."
6110 :    PRINT:PRINT
6120 :    PRINT"MOS :";T(1)"      Vous :"T(8)
6130 :    PRINT:PRINT:PRINT
6140 :    INPUT"Une autre Partie(O/N)";R$
6150 :
6160 :  RETURN
```

Vocabulaire

PRESENTATION

Parmi les quelques mots de la langue française, vous devez bien en connaître quelques milliers. Ce jeu fait appel :

- à votre connaissance actuelle de la langue française ;
- à votre mémoire à court terme.

Six initiales apparaissent sur l'écran. Vous avez une minute maximum pour indiquer un mot commençant par chacune de ces initiales. L'écran s'efface et les mêmes initiales réapparaissent. Vous devez alors entrer six mots différents, etc. Le jeu s'arrête quand la minute impartie est écoulée sans que vous ayez entré les six mots.

Si un mot a déjà été tapé, un message d'erreur apparaît et le mot n'est pas accepté.

ORGANIGRAMME DU JEU

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en :

- lecture du mot proposé ;
- test si le mot proposé a déjà été entré ou est incorrect ;
- affichage du temps écoulé depuis le début du jeu ;
- test si la temporisation maximum est atteinte ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et initialisation.
lignes 2000 à 2150 : jeu.
lignes 3000 à 3160 : fin du jeu.

Niveau 3 - **lignes 2200 à 2470** : test de validité de la réponse.

IDENTIFICATION DES VARIABLES

A : code ASCII des six lettres tirées au hasard.

A$: caractère saisi au clavier.

B$: mot saisi au clavier.

BIS : test si le mot est accepté.

CPT : compteur de temps écoulé.

I,J,K : index de boucles.

NJ : numéro de partie.

ME$: message d'erreur envoyé à l'écran.

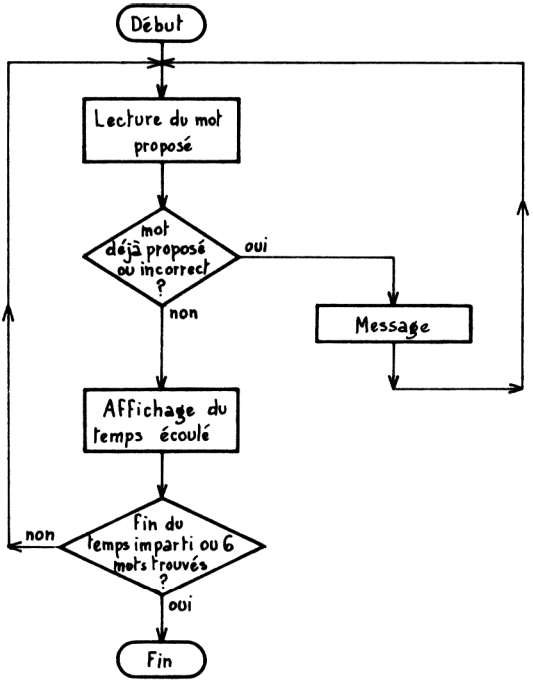
OK : test si le mot a déjà été entré.

R$: code de nouvelle partie.

T$: tableau des initiales tirées au hasard.

TJ$: tableau des mots entrés.

ORGANIGRAMME



```
10 :    REM VOCABULAIRE
20 :
30 :      GOSUB 1000 'Presentation et Initialisation
40 :      GOSUB 2000 'Jeu
50 :      GOSUB 3000 'Fin de jeu
60 :      IF R$="O" THEN 40 'Une autre Partie est en9a9
ee
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation et Initialisation
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    GOSUB 1600 'Initialisation
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation du jeu
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1150 :
1160 :      ATTRB 1,1 : COLOR 1 : CLS
1170 :      PRINT:PRINT"     VOCABULAIRE"
1180 :
1190 :    REM Objet
1200 :
1210 :      ATTRB 0,0 : PRINT:PRINT
1220 :      COLOR 4
1230 :      PRINT"OBJET : ";
1240 :      COLOR 6
1250 :      PRINT"Jeu de connaissances"
1255 :      PRINT"         et de memoire."
1260 :
1270 :    REM Job
1280 :
1290 :      PRINT:PRINT
1300 :      COLOR 5
1310 :      PRINT"JOB    : ";
1320 :      COLOR 6
1330 :      PRINT"MO5 va afficher 6 initiales"
1340 :      PRINT"         tirees au hasard. Vous devrez
"
1350 :      PRINT"         trouver un mot Pour cha9ue"
1360 :      PRINT"         initiale en moins d'1 minute.
"
```

```
1370 :      PRINT"         Cela fait, l'ecran s'effacera
 "
1380 :      PRINT"         et la meme demande sera faite
."
1390 :      PRINT"         Le jeu s'arretera quand les m
ots";
1400 :      PRINT"         n'auront pas ete trouves dans
 le";
1410 :      PRINT"         temps imparti..."
1420 :
1430 :    REM Debut
1440 :
1450 :      PRINT:PRINT
1460 :      COLOR 3
1470 :      PRINT"DEBUT : ";
1480 :      COLOR 6
1490 :      PRINT"Pour commencer, appuyez sur"
1500 :      PRINT"         une touche quelconque."
1510 :
1520 :      A=RND : A$=INKEY$ : IF A$="" THEN 1520 'Att
ente de l'appui
1530 :
1540 :    REM Dimentionnement du tableau de jeu
1550 :
1560 :      DIM TJ(40,6) : NJ=1
1570 :
1580 : RETURN
1590 : REM ****************************************
1600 : REM Initialisation du jeu
1610 :
1660 :    REM Affichage Preliminaire
1670 :
1680 :      CLS : ATTRB 1,1 : COLOR 1
1690 :      PRINT : PRINT"     VOCABULAIRE"
1700 :      COLOR 7,4 : ATTRB 0,0
1710 :
1720 :    REM Tirage au sort des 6 initiales et afficha
ge
1730 :
1740 :      FOR I=1 TO 6
1750 :        A=INT(RND*26)+1 : OK=-1
1760 :        IF I=1 THEN 1780
1761 :        FOR J=1 TO I-1
1762 :          IF T(J)=A THEN OK=1
1763 :        NEXT J
1770 :        IF OK=1 THEN 1750 'Lettre deja tiree
1780 :        PRINT CHR$(A+64) : PRINT : PRINT : T(I)=A
```

```
1790 :      NEXT I
1800 :
1810 : RETURN
1820 : REM ******************************************
2000 : REM Jeu
2010 :
2020 :   COLOR 3,0 : LOCATE 10,20,1 : PLAY"SIPSI"
2030 :   PRINT"A vous de Jouer..."
2031 :   FOR I=1 TO 500 : NEXT I
2032 :   LOCATE 10,20 : PRINT SPC(40)
2035 :   COLOR 6
2036 :   LOCATE 13,23 : PRINT"Temps :"0
2040 :
2050 :   FOR I=1 TO 6
2060 :     LOCATE 3,3*I+1 : B$="" : X=3 : Y=3*I+1
2070 :     A$=INKEY$ : B$=B$+A$
2080 :     CPT=CPT+1 'Compteur de temps ecoule
2090 :     IF A$<>"" THEN IF ASC(A$)=13 THEN GOSUB 220
0 'RePonse OK?
2095 :     IF B=1 THEN B=0 : IF BIS=-1 THEN 2120 ELSE
2060
2096 :     IF A$<> "" THEN IF ASC(A$)<>13 THEN X=X+1 :
 PRINT A$;
2100 :     IF INT(CPT/50)=CPT/50 THEN LOCATE 13,23 : P
RINT "Temps :"CPT/50 : LOCATE X,Y
2110 :     IF CPT/50 <= 12 AND I<>7 THEN 2070
2120 :   NEXT I
2130 :
2140 : RETURN
2150 : REM ******************************************
2200 : REM Test de la validite de la reponse
2210 :
2220 :   REM Test No 1
2230 :   REM Le mot commence-t-il Par l'initiale speci
fiee
2240 :
2250 :     ME$="OK"
2260 :     IF LEFT$(B$,1)<>CHR$(T(I)+64) THEN ME$="Mau
vaise initiale !"
2270 :
2280 :   REM Test No 2
2290 :   REM Le mot a-t-il deja ete ProPose
2300 :
2310 :     OK=-1
2320 :     IF NJ=0 THEN 2400 'Memorisation
2330 :     FOR K=1 TO NJ
2360 :       IF TJ$(K,I)=B$ THEN OK=1
2370 :     NEXT K
```

```
2380 :      IF OK=1 THEN ME$="Mot deja entre !"
2390 :
2400 :    REM Memorisation du mot ProPose
2410 :
2420 :      TJ$(NJ,I)=B$ : B=1
2430 :      LOCATE 20,I*3+1 : PRINT ME$;SPC(20)
2440 :      IF ME$="OK" THEN BIS=-1 ELSE BIS=1
2450 :
2460 : RETURN
2470 : REM ******************************************
3000 : REM Fin de la Partie
3010 :
3020 :    LOCATE 0,21 : NJ=NJ+1
3030 :    IF CPT/50>=12 THEN PRINT"LouPe" ELSE PRINT"Ga
9ne"
3040 :
3050 :    PRINT"Un autre essai (O/N)" : R$=INPUT$(1)
3055 :    IF R$="N" THEN CLS
3060 :
3070 :    REM Les mots deja entres sont effaces
3080 :
3090 :      FOR I=1 TO 6
3100 :        LOCATE 3,I*3+1
3110 :        PRINT SPC(40)
3120 :      NEXT I
3130 :
3140 :      LOCATE 0,21 : PRINT SPC(80) : CPT=0
3150 :
3160 : RETURN
```

GRAPHIQUES ET MATHÉMATIQUES

Tracé de courbes
y = f (x)

PRESENTATION

Ce petit programme peut aider tous les étudiants qui font des études de courbes. En effet, après l'entrée de l'équation de la courbe à représenter et du domaine d'étude qui omet les éventuels points aberrants, MO5 représente la courbe en haute résolution.

ORGANIGRAMME DU PROGRAMME

Le programme possède la structure hiérarchique à deux niveaux définie en annexe.

Le module de traitement se décompose en six actions élémentaires :

- saisie de l'équation de la courbe ;
- saisie du domaine d'étude ;
- recherche des pas Ox et Oy ;
- tracé ;
- informations sur la courbe affichées à l'écran ;
- test si agrandissement demandé ;

d'où l'organigramme ci-contre.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 70** : module de commande.

Niveau 2 - **lignes 1000 à 1430** : présentation.
lignes 2000 à 2720 : saisie de l'équation et tracé.

IDENTIFICATION DES VARIABLES

A$: attente d'un caractère pour passer à la suite.

R$: test si agrandissement demandé.

X : abcisse courbe.

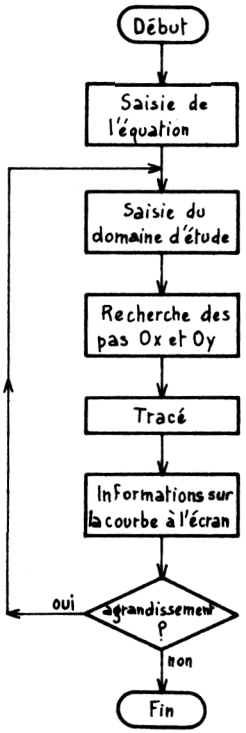
X1 : borne minimum du domaine d'étude.

X2 : borne maximum du domaine d'étude.

XPAS : pas en X.

YPAS : pas en Y.

ORGANIGRAMME



```
10 :    REM TRACE DE COURBES Y = f(x)
20 :
30 :      GOSUB 1000 'Presentation
40 :      GOSUB 2000 'Trace
50 :
60 : RETURN
70 : REM ****************************************
1000 : REM Presentation
1010 :
1020 :    SCREEN 6,0,0 : LOCATE 1,1,0
1030 :
1040 :    REM Titre
1050 :
1060 :      CLS : COLOR 1 : ATTRB 0,1
1070 :      PRINT : PRINT"        TRACE DE COURBES Y=F(x
)"
1080 :
1090 :    REM Objet
1100 :
1110 :      ATTRB 0,0 : PRINT:PRINT
1120 :      COLOR 4
1130 :      PRINT"OBJET : ";
1140 :      COLOR 6
1150 :      PRINT"Representation graphique de"
1155 :      PRINT"         courbes a une variable."
1160 :
1170 :    REM Job
1180 :
1190 :      PRINT:PRINT:PRINT
1200 :      COLOR 5
1210 :      PRINT"JOB    : ";
1220 :      COLOR 6
1230 :      PRINT"Apres avoir entre"
1240 :      PRINT"         l'equation et le domaine d'et
ude";
1250 :      PRINT"         de la courbe, MO5 effectuera
son";
1260 :      PRINT"         trace en haute resolution."
1270 :      PRINT"         Il sera alors possible"
1280 :      PRINT"         d'obtenir un 'agrandissement'
"
1290 :      PRINT"         de la courbe en restreignant"
1300 :      PRINT"         son domaine d'etude."
1310 :
1320 :    REM Debut
1330 :
1340 :      PRINT:PRINT:PRINT
1350 :      COLOR 3
```

```
1360 :      PRINT"DEBUT : ";
1370 :      COLOR 6
1380 :      PRINT"Pour commencer, appuyez sur"
1390 :      PRINT"          une touche quelconque."
1400 :      A$=INPUT$(1) 'Attente de l'appui
1410 :
1420 :  RETURN
1430 :  REM ****************************************
2000 :  REM Saisie de l'equation et trace
2010 :
2020 :    CLS : COLOR 1
2030 :
2040 :    REM Saisie de l'equation
2050 :
2060 :      PRINT"          SAISIE DE L'EQUATION :"
2070 :      LOCATE 3,8 : COLOR 3
2080 :      PRINT"Vous allez entrer l'equation"
2090 :      PRINT"   sous la forme :" : PRINT
2100 :      PRINT"   2230 DEF FNA(X) = SIN(X)"
2110 :      PRINT:PRINT:PRINT:PRINT"   Vous taperez ens
uite :"
2120 :      PRINT:PRINT"   RUN 2200"
2130 :      COLOR 6 : LOCATE 0,18,1
2140 :
2150 :      STOP
2155 :
2160 :    REM L'equation est saisie
2170 :    REM Saisie du domaine de definition
2180 :
2200 :      CLS : COLOR 1
2210 :      PRINT"    SAISIE DU DOMAINE DE DEFINITION"
2220 :
2230 DEF FNA(X) = SIN(X)
2260 :      LOCATE 4,8 : COLOR 3
2270 :      PRINT"Vous allez entrer les valeurs"
2280 :      PRINT"   minimale Xmin et maximale Xmax"
2290 :      PRINT"   du domaine d'etude :"
2300 :      PRINT:PRINT:PRINT
2310 :      INPUT"           Xmin = ";X1
2320 :      INPUT"           Xmax = ";X2
2330 :      COLOR 6 : LOCATE 0,18,0 'Plus de curseur
2340 :      PRINT"    Un moment s'il vous plait,"
2350 :      PRINT"    je suis en train de calculer..."
2360 :
2370 :    REM Le domaine d'etude est entre
2380 :
2390 :      XPAS=(X2-X1)/320 'Pas en abcisse
2400 :
2410 :    REM Recherche du pas en Ordonnee
```

```
2420 :
2430 :      Y1=1E33 : Y2=-1E33 'Initialisation du min e
t du max en ordonnee
2440 :
2450 :      FOR X=X1 TO X2 STEP XPAS
2460 :        A=FNA(X)
2470 :        IF A<Y1 THEN Y1=A : X3=X
2480 :        IF A>Y2 THEN Y2=A : X4=X
2490 :      NEXT X
2500 :
2510 :      YPAS=(Y1-Y2)/160
2520 :
2530 :    REM Trace
2540 :
2550 :      CLS : COLOR 6
2560 :
2570 :      FOR X=X1 TO X2 STEP XPAS
2580 :        Y=FNA(X)
2590 :        PSET(INT((1/XPAS)*(X-X1)),INT((1/YPAS)*(Y
-Y2))),6
2600 :      NEXT X
2610 :
2620 :      LOCATE 0,20
2630 :      PRINT"Domaine d'etude : ("X1","X2")"
2640 :      PRINT"Y max en X="INT(X4*100)/100",Y min en
 X="INT(X3*100)/100
2650 :      PRINT"Y max="INT(Y2*100)/100"   Ymin="INT(Y
1*100)/100
2660 :
2670 :      COLOR 3 : PRINT"Un a9randissement (O/N)"
2680 :      R$=INPUT$(1)
2690 :
2700 :      IF R$="O" THEN CLEAR : GOTO 2200 'A9randiss
ement
2710 :
2720 :      COLOR 6 : LOCATE 1,1,1 : CLS
```

Tracé de courbes y (t), x (t)

PRESENTATION

Ce programme peut être employé dans deux voies différentes :

- comme aide mathématique (représentation instantannée d'une courbe par l'entrée de ses équations) ;
- comme générateur de motifs mathématiques graphiques.

La première voie, si elle n'évite pas à l'étudiant les fastidieuses recherches de tangentes, points d'inflexion et assymptotes nécessaires à tout "bon" tracé de courbe, permet cependant de concrétiser rapidement une impression plus ou moins juste sur l'allure de la courbe à étudier.

La seconde voie permet d'obtenir d'assez jolis dessins par la seule entrée d'équations (dont quelques exemples sont donnés par la suite) et du domaine d'étude.

ORGANIGRAMME DU PROGRAMME

Le programme possède la structure hiérarchique à deux niveaux définie en annexe.

Le module de traitement se décompose en six actions élémentaires :

- saisie des équations X = f(t), Y = g(t) ;
- saisie du domaine d'étude (T min, T max) ;
- recherche des pas en X et en Y ;
- tracé ;
- informations sur la courbe ;
- test si agrandissement demandé ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 70** : module de commande.

Niveau 2 - **lignes 1000 à 1430** : présentation.
lignes 2000 à 2720 : saisie des équations et tracé.

IDENTIFICATION DES VARIABLES

A$: attente d'un caractère au clavier.

R$: test si agrandissement demandé.

TPAS : pas de progression en T.

X : abcisse de la courbe.

X1 : borne min de X(t).

X2 : borne max de X(t).

XPAS : pas en X.

YPAS : pas en Y.

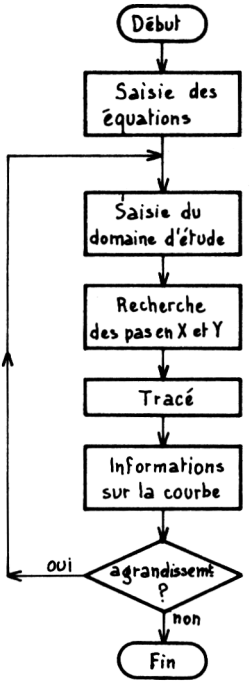
Y1 : borne min de Y(t).

Y2 : borne max de Y(t).

Exemples d'équations

Les exemples donnés dans le programme "Tracé de courbes Y(t), X(t) imbriquées" s'appliquent également à ce programme.

ORGANIGRAMME



```
10 :   REM TRACE DE COURBES Y(t),X(t)
20 :
30 :     GOSUB 1000 'Presentation
40 :     GOSUB 2000 'Trace
50 :
60 : RETURN
70 : REM ****************************************
1000 : REM Presentation
1010 :
1020 :   SCREEN 6,0,0 : LOCATE 1,1,0
1030 :
1040 :   REM Titre
1050 :
1060 :     CLS : COLOR 1 : ATTRB 0,1
1070 :     PRINT : PRINT"       TRACE DE COURBES Y(t),
X(t)"
1080 :
1090 :   REM Objet
1100 :
1110 :     ATTRB 0,0 : PRINT:PRINT
1120 :     COLOR 4
1130 :     PRINT"OBJET : ";
1140 :     COLOR 6
1150 :     PRINT"Representation graphique de"
1155 :     PRINT"         courbes Y(t),X(t)."
1160 :
1170 :   REM Job
1180 :
1190 :     PRINT:PRINT:PRINT
1200 :     COLOR 5
1210 :     PRINT"JOB    : ";
1220 :     COLOR 6
1230 :     PRINT"Apres avoir entre les"
1240 :     PRINT"         equations et le domaine d'etu
de";
1250 :     PRINT"         de la courbe, MO5 effectuera
son";
1260 :     PRINT"         trace en haute resolution."
1270 :     PRINT"         Il sera alors possible"
1280 :     PRINT"         d'obtenir un 'agrandissement'
"
1290 :     PRINT"         de la courbe en restreignant"
1300 :     PRINT"         son domaine d'etude."
1310 :
1320 :   REM Debut
1330 :
1340 :     PRINT:PRINT:PRINT
1350 :     COLOR 3
```

```
1360 :      PRINT"DEBUT : ";
1370 :      COLOR 6
1380 :      PRINT"Pour commencer, appuyez sur"
1390 :      PRINT"         une touche quelconque."
1400 :      A$=INPUT$(1) 'Attente de l'appui
1410 :
1420 :  RETURN
1430 :  REM ****************************************
2000 :  REM Saisie de l'equation et trace
2010 :
2020 :    CLS : COLOR 1
2030 :
2040 :    REM Saisie de l'equation
2050 :
2060 :      PRINT"         SAISIE DE L'EQUATION :"
2070 :      LOCATE 3,6 : COLOR 3
2080 :      PRINT"Vous allez entrer les equations"
2090 :      PRINT"   sous la forme :" : PRINT
2100 :      PRINT"   2230 DEF FNA(T) = SIN(T)"
2101 :      PRINT"   2231 DEF FNB(T) = COS(T)"
2102 :      PRINT : PRINT"   respectivement pour les eq
uations"
2103 :      PRINT"   X(t) et Y(t)."
2110 :      PRINT:PRINT:PRINT"   Vous taperez ensuite :
"
2120 :      PRINT:PRINT"   RUN 2200"
2130 :      COLOR 6 : LOCATE 0,19,1
2140 :
2150 :      STOP
2155 :
2160 :    REM L'equation est saisie
2170 :    REM Saisie du domaine de definition
2180 :
2200 :      CLS : COLOR 1
2210 :      PRINT"    SAISIE DU DOMAINE DE DEFINITION"
2220 :
2230 DEF FNA(X) = SIN(X)
2231 DEF FNB(T) = COS(T)
2260 :      LOCATE 4,8 : COLOR 3
2270 :      PRINT"Vous allez entrer les valeurs"
2280 :      PRINT"   minimale Tmin et maximale Tmax"
2290 :      PRINT"   du domaine d'etude :"
2300 :      PRINT:PRINT:PRINT
2310 :      INPUT"           Tmin = ";T1
2320 :      INPUT"           Tmax = ";T2
2330 :      COLOR 6 : LOCATE 0,18,0 'Plus de curseur
2340 :      PRINT"    Un moment s'il vous plait,"
2350 :      PRINT"    je suis en train de calculer..."
```

```
2360 :
2370 :    REM Le domaine d'etude est entre
2380 :
2390 :      TPAS=(T2-T1)/320 'Pas en abcisse
2400 :
2410 :    REM Recherche des Pas en X et en Y
2420 :
2430 :      X1=1E33 : X2=-1E33 : Y1=1E33 : Y2=-1E33
2440 :
2450 :      FOR T=T1 TO T2 STEP TPAS
2460 :        A=FNA(T) : B=FNB(T)
2470 :        IF A<X1 THEN X1=A : X3=T
2475 :        IF A>X2 THEN X2=A : X4=T
2480 :        IF B<Y1 THEN Y1=B : Y3=T
2485 :        IF B>Y2 THEN Y2=B : Y4=T
2490 :      NEXT T
2500 :
2510 :      XPAS=(X1-X2)/320 : YPAS=(Y1-Y2)/160
2520 :
2530 :    REM Trace
2540 :
2550 :      CLS : COLOR 6
2560 :
2570 :      FOR T=T1 TO T2 STEP TPAS
2580 :        X=FNA(T) : Y=FNB(T)
2590 :        PSET(INT((1/XPAS)*(X1-X)),INT((1/YPAS)*(Y
-Y2))),6
2600 :      NEXT T
2610 :
2620 :      LOCATE 0,20
2630 :      PRINT"Domaine d'etude : ("T1","T2")"
2640 :      PRINT"Y max en X="INT(X4*100)/100",Y min en
 X="INT(X3*100)/100
2650 :      PRINT"Y max="INT(Y2*100)/100"   Ymin="INT(Y
1*100)/100
2660 :
2670 :      COLOR 3 : PRINT"Un agrandissement (O/N)"
2680 :      R$=INPUT$(1)
2690 :
2700 :      IF R$="O" THEN CLEAR : GOTO 2200 'Agrandiss
ement
2710 :
2720 :      COLOR 6 : LOCATE 1,1,1 : CLS
```

Tracé de courbes y (t), x (t) imbriquées

PRESENTATION

Ce programme s'adresse à tous ceux qui aiment les belles représentations de courbes mathématiques.

C'est une variante du programme précédent (*Tracé de courbes Y(t), X(t)*). Il permet d'imbriquer autant de courbe de même équation qu'on le désire, ce qui donne une "profondeur" à la représentation...

Seulement quelques lignes ont été changées par rapport à "Tracé de courbes Y(t),X(t) :

- dans la saisie du domaine d'étude : **lignes 2270 à 2390** ;
- dans le tracé de la courbe : **lignes 2450 à 2610.**

ORGANIGRAMME DU PROGRAMME

Il est identique au précédent :

- structure hiérarchique à deux niveaux ;
- même découpage modulaire.

IDENTIFICATION DES VARIABLES

Mêmes variables que pour le programme "Tracé de courbes Y(t), X(t)" avec en plus :

NC : nombre de courbes imbriquées.

NP : nombre de points par courbe.

Exemples d'équations

Essayez donc diverses combinaisons de fonctions trigonométriques. Les couples d'équations suivants donnent de bons résultats avec 150 points et un domaine d'étude compris entre 0 et 65.

$$\begin{cases} (1 + \cos(T)) * \sin(T) \\ (1 + \cos(T)) * \cos(T) \end{cases} \qquad \begin{cases} (1 - \sin(T)) * \cos(T) \\ (1 + \cos(T)) * \cos(T) \end{cases}$$

$$\begin{cases} (1 + \cos(T)) * \cos(T) \\ (1 + \cos(T)) * \sin(T) \end{cases} \qquad \begin{cases} (1 - \cos(2*T)) * \sin(T) \\ (1 - \cos(T)) * \cos(T) \end{cases}$$

$$\begin{cases} (1 + \cos(2*T)) * \sin(2*T) \\ (1 + \cos(T)) * \cos(T) \end{cases} \qquad \begin{cases} (1 + \cos(2*T)) * \cos(2*T) \\ (1 + \sin(T/2))* \sin(T) \end{cases}$$

$$\begin{cases} (1 + \sin(2*T)) * \cos(T) \\ (1 + \sin(2*T)) * \sin(T) \end{cases}$$

etc.

```
15 :   REM IMBRIQUEES
2290 :     PRINT"   du domaine d'etude, Puis"
2291 :     PRINT"   le nombre de cardioides imbriquees
"
2292 :     PRINT"   NC, et le nombre de Points Par"
2293 :     PRINT"   cardioide NP."
2321 :     INPUT"              NC  = ";NC
2322 :     INPUT"              NP  = ";NP
2330 :     COLOR 6 : LOCATE 0,22,0 'Plus de curseur
2390 :     TPAS=(T2-T1)/NP 'Pas en abcisse
2460 :       A=FNA(T)*NC : B=FNB(T)*NC
2565 :     FOR K=1 TO NC
2580 :       X=FNA(T)*K : Y=FNB(T)*K
2605 :     NEXT K
```

Renverse

PRESENTATION

Ce jeu de réflexion demande une certaine pratique pour être optimalement utilisé. Il met en jeu votre faculté de "vision dans l'espace" et votre "mémoire à court terme".

Le jeu se compose d'une suite de neuf chiffres (de 1 à 9) écrits dans le désordre. En un minimum de coups, vous devez construire une suite croissante par retournements successifs. Il est assez facile d'arriver à la suite demandée en un grand nombre de coups, mais essayez donc de diminuer d'un le nombre maximum de retournements à chaque partie gagnée, et vous verrez très vite que la tâche n'est plus si simple...

ORGANIGRAMME DU PROGRAMME

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en cinq actions élémentaires :

- tirage aléatoire des neuf chiffres à afficher ;
- affichage des neuf chiffres ;
- saisie du nombre à retourner ;
- retournement ;
- test si fin de partie ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 90** : module de commande.

Niveau 2 - **lignes 1000 à 1060** : présentation et exemple.
lignes 2000 à 2480 : jeu.
lignes 3000 à 3080 : fin de partie.

Niveau 3 - **lignes 1100 à 1530** : présentation du jeu.
lignes 1600 à 1850 : exemple.

IDENTIFICATION DES VARIABLES

A : chiffre tiré au hasard.

A$: attente d'un caractère au clavier.

BP : nombre de chiffres bien placés.

I,J : index de boucles.

NC : nombre de coups joués.

OK : test si nombre déjà tiré au hasard.

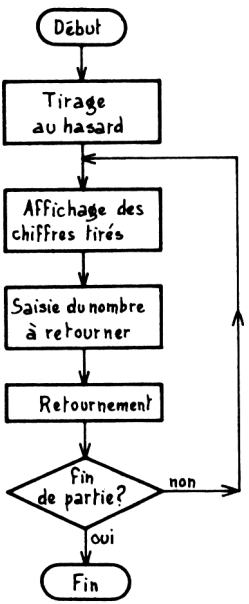
R : nombre de chiffres à retourner.

R$: validation d'une nouvelle partie.

T : tableau de jeu.

TP : tableau dans lequel se fait le retournement.

ORGANIGRAMME



```
10 :    REM RENVERSE
20 :
30 :      GOSUB 1000 'Presentation et Exemple
40 :      GOSUB 2000 'Jeu
50 :      GOSUB 3000 'Fin de la Partie
60 :      IF R$="O" THEN 40 'Une autre Partie est en9a9
ee
70 :
80 :    END
90 :    REM ****************************************
1000 :  REM Presentation et Exemple
1010 :
1020 :    GOSUB 1100 'Presentation
1030 :    IF A$="N" THEN GOSUB 1600 'Exemple
1040 :
1050 :  RETURN
1060 :  REM ****************************************
1100 :  REM Presentation du Jeu
1110 :
1120 :    SCREEN 6,0,0 : LOCATE 1,1,0
1130 :
1140 :    REM Titre
1150 :
1160 :      ATTRB 1,1 : COLOR 1 : CLS
1170 :      PRINT:PRINT"      RENVERSE"
1180 :
1190 :    REM Objet
1200 :
1210 :      ATTRB 0,0 : PRINT:PRINT:PRINT
1220 :      COLOR 4
1230 :      PRINT"OBJET : ";
1240 :      COLOR 6
1250 :      PRINT"Jeu de reflexion."
1260 :
1270 :    REM Job
1280 :
1290 :      PRINT:PRINT:PRINT
1300 :      COLOR 5
1310 :      PRINT"JOB   : ";
1320 :      COLOR 6
1330 :      PRINT"9 chiffres vont aPParaitre a"
1340 :      PRINT"         l'ecran. Par retournements"
1350 :      PRINT"         successifs, vous devrez"
1360 :      PRINT"         restituer une suite croissant
e."
1370 :      PRINT"         Si vous demandez un retournem
ent";
```

→

```
1380 :      PRINT"         de N, les N chiffres de gauch
e"
1390 :      PRINT"         seront retournes."
1400 :
1410 :    REM Debut
1420 :
1430 :      PRINT:PRINT:PRINT
1440 :      COLOR 3
1450 :      PRINT"DEBUT : ";
1460 :      COLOR 6
1470 :      PRINT"Si vous avez compris, tapez <O>,";
1480 :      PRINT"         sinon, tapez <N>."
1490 :
1500 :      A=RND : A$=INKEY$ : IF A$="" THEN 1500 'Att
ente de l'appui
1510 :
1520 :  RETURN
1530 :  REM ****************************************
1600 :  REM Exemple
1610 :
1620 :    CLS : ATTRB 1,1 : COLOR 1
1630 :    PRINT:PRINT"     RENVERSE"
1640 :
1650 :    ATTRB 0,0 : COLOR 6
1660 :    PRINT:PRINT
1670 :    PRINT"Cet exemple va servir a vous fixer les"
1680 :    PRINT"idees sur la regle du jeu."
1690 :    PRINT
1700 :    COLOR 3 : PRINT"   Voici une suite de 9 chiff
res:"
1710 :    ATTRB 1,1 : PRINT : PRINT" 4 3 2 1 5 6 7 8 9"
  : ATTRB 0,0
1720 :    PRINT:PRINT"En un seul retournement, vous dev
ez"
1730 :    PRINT"constituer une suite croissante."
1740 :    COLOR 6 : PRINT:PRINT
1750 :    INPUT"Combien dois-je en retourner ";R
1760 :    COLOR 1
1770 :    IF R<>4 THEN PRINT"Il fallait en retourner 4.
" ELSE PRINT"OK."
1780 :
1790 :    PRINT : COLOR 6
1800 :    PRINT"Pressez une touche pour commencer le je
u"
1805 :    LOCATE 1,1,0 'Le curseur disparait
1810 :
1820 :    A$=INPUT$(1) 'Attente de l'appui
```

```
1830 :
1840 : RETURN
1850 : REM ****************************************
2000 : REM Jeu
2010 :
2020 :   REM Tirage aleatoire des 9 chiffres a affiche
r
2030 :
2040 :     FOR I=1 TO 9
2050 :       A=INT(RND*9)+1 : OK=-1
2060 :       IF I=1 THEN 2100 'Double impossible
2070 :       FOR J=1 TO I-1
2080 :         IF T(J)=A THEN OK=1 'Nombre deja tire
2090 :       NEXT J
2100 :       IF OK=1 THEN 2050 ELSE T(I)=A 'Retirage o
u memorisation
2110 :     NEXT I
2120 :
2130 :   REM Boucle Principale
2140 :
2150 :     CLS : ATTRB 1,1 : COLOR 1
2160 :     PRINT:PRINT"       RENVERSE"
2170 :
2180 :     LOCATE 3,10 : COLOR 3
2190 :     FOR I=1 TO 9
2200 :       LOCATE (I-1)*4+2,8 : PRINT T(I)
2210 :     NEXT I
2220 :
2230 :     ATTRB 0,0 : COLOR 6
2240 :     LOCATE 5,18
2250 :     INPUT"Combien dois-je en retourner ";R
2260 :
2270 :     REM Retournement
2280 :
2290 :       FOR I=1 TO R
2300 :         TP(I)=T(R-I+1)
2310 :       NEXT I
2320 :
2330 :       FOR I=1 TO R
2340 :         T(I)=TP(I)
2350 :       NEXT I
2360 :
2370 :     NC=NC+1 'Nombre de coups joues
2380 :
2390 :     REM Test de fin de Partie
2400 :
2405 :       BP=0 'RAZ de l'indicateur de chiffre bien
 Place
```

```
2410 :        FOR I=1 TO 9
2420 :          IF T(I)=I THEN BP=BP+1
2430 :        NEXT I
2440 :
2450 :      IF BP<>9 THEN 2130 'Le jeu n'est pas fini
2460 :
2470 : RETURN
2480 : REM ****************************************
3000 : REM Fin de la Partie
3010 :
3020 :   CLS
3030 :   PRINT"Suite croissante etablie en"NC"coups."
3040 :
3050 :   LOCATE 0,8
3060 :   INPUT"Une autre Partie (O/N) ";R$
3070 :
3080 : RETURN
```

UTILITAIRES

Palette de couleurs

PRESENTATION

Si l'on rapproche petit à petit deux points élémentaires de couleurs différentes, il arrive un moment ou l'oeil ne fait plus la différence entre les deux points : il perçoit un seul point dont la couleur est le mélange de la couleur de chaque point.

Ce principe est utilisé ici : la superposition d'un fond de couleur A et d'un motif de couleur B arrive à tromper l'oeil qui perçoit une couleur C différente de A et de B, si le motif est bien choisi.

TO7 possède huit couleurs ; MO5 possède seize couleurs.

Ce programme vous propose d'étendre le nombre de couleurs à :

- 64 pour TO7 ;
- 256 pour MO5.

ORGANIGRAMME DE L'UTILITAIRE

Le programme possède la structure hiérarchique à deux niveaux définie en annexe.

Le module de traitement se décompose en deux actions élémentaires :

- affichage des pastels correspondant à la couleur de fond choisie ;
- attente de l'appui sur une touche pour changer de couleur de fond ;

d'où l'organigramme ci-contre.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 95** : définition du motif graphique et affichage.

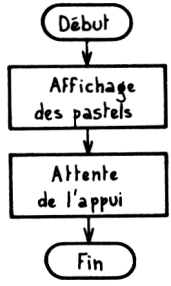
Niveau 2 - **lignes 1000 à 1060** : définition du motif graphique.
lignes 2000 à 2220 : affichage d'une série de pastels.

IDENTIFICATION DES VARIABLES

A$: attente d'une action au clavier.

I,J : index de boucles.

ORGANIGRAMME



```
10 :   REM PALETTE DE COULEURS TO7
20 :
30 :     CLEAR,,1 'Declaration d'un motif 9raPhi9ue
40 :     GOSUB 1000 'Definition du motif 9raPhi9ue
50 :
60 :     FOR I=0 TO 7
70 :       GOSUB 2000 'Afficha9e d'une serie de Pastel
s
80 :     NEXT I
85 :     LOCATE 1,1,1 : CLS 'Curseur aPParent
90 :
94 :   END
95 : REM ****************************************
1000 : REM Definition du motif 9raPhi9ue
1010 :
1020 :   SCREEN 6,0,0
1030 :   DEFGR$(0)=85,170,85,170,85,170,85,170
1040 :
1050 : RETURN
1060 : REM ****************************************
2000 : REM Afficha9e d'une serie de Pastels
2010 :
2020 :   CLS : ATTRB 1,1 : COLOR 1
2030 :   PRINT:PRINT"PALETTE DE COULEURS"
2040 :
2050 :   ATTRB 0,0 : LOCATE 0,5
2055 :   PRINT"Trame   Fond"
2060 :
2070 :   FOR J=0 TO 7
2075 :     LOCATE 2,J*2+6
2080 :     COLOR J,0 : PRINT GR$(0);
2090 :     COLOR 6,0 : PRINT"   +   ";
2100 :     COLOR J,I : PRINT " ";
2110 :     COLOR 6,0 : PRINT"   =  ";
2120 :     COLOR J,I : PRINT GR$(0);
2130 :     COLOR 6,0 : PRINT"     : COLOR"J","I
2140 :   NEXT J
2150 :
2160 :   PRINT : PRINT : PRINT"APPuyez sur une touche
Pour continuer."
2170 :   LOCATE 1,1,0 'Le curseur disParait
2180 :
2190 :   A$=INPUT$(1) 'Attente de l'aPPui
2200 :
2210 : RETURN
2220 : REM ****************************************
```

Gestion de fichiers

PRESENTATION

Lorsque vous débranchez l'ordinateur, tout ce qui se trouvait en mémoire vive est perdu. Si vous désirez garder les données qui se trouvaient en mémoire vive, un moyen simple consiste à construire un fichier de données et à le stocker sur support magnétique (bande, disquette ou disque dur).

Ce programme gère des fichiers de données sur disquettes. Il réalise les opérations suivantes :

- création de structure ;
- création de fiche possédant la structure définie ;
- visualisation de l'ensemble des fiches ;
- modification d'une fiche ;
- suppression d'une fiche ;
- lecture d'un fichier de données sur disquette ;
- écriture d'un fichier de données sur disquette.

ORGANIGRAMME DE L'UTILITAIRE

Le programme possède la structure hiérarchique à trois niveaux définie en annexe.

Le module de traitement se décompose en trois actions élémentaires :

- choix d'une action dans le menu ;
- action ;
- retour au menu ou fin ;

d'où l'organigramme ci-après.

STRUCTURE DU PROGRAMME

Niveau 1 - **lignes 10 à 70** : module de commande.

Niveau 2 - **lignes 1000 à 1430** : présentation.
lignes 2000 à 2270 : menu principal.

Niveau 3 - **lignes 3000 à 3250** : création de structure.
lignes 3500 à 3720 : création de fiche.
lignes 4000 à 4170 : visualisation des fiches.
lignes 4500 à 4810 : modification d'une fiche.
lignes 5000 à 5310 : suppression d'une fiche.
lignes 5500 à 5870 : lecture disquette.
lignes 6000 à 6320 : écriture disquette.
lignes 6500 à 6660 : fin de programme.
lignes 9000 à 9080 : message de fin d'action.

IDENTIFICATION DES VARIABLES

A$: attente d'une action clavier.

A$: image du fichier en mémoire.

B,B$: choix d'une des actions du menu.

C$: attente d'une action clavier dans le module de fin d'action.

FIN : indicateur de fin de programme.

I,J : index de boucle.

LA$: tableau des labels.

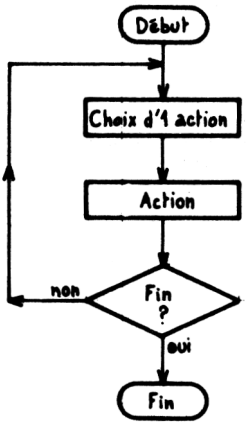
NA : nombre d'articles par fiche.

NF : nombre de fiches en mémoire.

NF$: nom du fichier à lire ou à écrire sur disquette.

R$: décision de poursuivre ou d'arrêter le module en cours.

ORGANIGRAMME




```
10 :    REM GESTION DE FICHIERS
20 :
30 :      GOSUB 1000 'Presentation
40 :      GOSUB 2000 'Menu Principal
50 :
60 : END
70 : REM ****************************************
1000 : REM Presentation
1010 :
1020 :    SCREEN 6,0,0 : LOCATE 1,1,0
1030 :
1040 :    REM Titre
1050 :
1060 :      CLS : COLOR 1 : ATTRB 1,1
1070 :      PRINT : PRINT"GESTION DE FICHIERS"
1080 :
1090 :    REM Objet
1100 :
1110 :      ATTRB 0,0 : PRINT:PRINT
1120 :      COLOR 4
1130 :      PRINT"OBJET : ";
1140 :      COLOR 6
1150 :      PRINT"Utilitaire de Gestion"
```

```
1155 :      PRINT"         de fichiers."
1160 :
1170 :    REM Job
1180 :
1190 :      PRINT:PRINT:PRINT
1200 :      COLOR 5
1210 :      PRINT"JOB    : ";
1220 :      COLOR 6
1230 :      PRINT"Ce Pro9ramme Permet de lire"
1240 :      PRINT"         et de stocker des donnees sur"
1250 :      PRINT"         dis9uette. Vous Pouvez e9ale
                                                    ment";
1260 :      PRINT"         creer la structure de votre"
1270 :      PRINT"         fichier, creer des fiches"
1280 :      PRINT"         Possedant la structure"
1290 :      PRINT"         Precedemment definie, modifier"
1300 :      PRINT"         une fiche ou la suPPrimer."
1310 :
1320 :    REM Debut
1330 :
1340 :      PRINT:PRINT:PRINT
1350 :      COLOR 3
1360 :      PRINT"DEBUT  : ";
1370 :      COLOR 6
1380 :      PRINT"Pour commencer, aPPuyez sur"
1390 :      PRINT"         une touche 9uelcon9ue."
1400 :      A$=INPUT$(1) 'Attente de l'aPPui
1405 :      FIN=1 'Indicateur de fin de Pro9ramme
1406 :      DIM A$(20,100),LA$(20) '100 Fiche max, 20
article/fiche max.
1410 :
1420 :  RETURN
1430 :  REM ****************************************
2000 :  REM Menu PrinciPal
2010 :
2020 :    REM Titre
2030 :
2040 :      CLS : ATTRB 1,1 : COLOR 1
2050 :      PRINT:PRINT"GESTION DE FICHIERS"
2060 :
2070 :    REM Menu
2080 :
2090 :      ATTRB 0,0 : COLOR 6 : LOCATE 0,6
2100 :      PRINT"Vous Pouvez :" : PRINT : PRINT
2110 :      PRINT"1 - Creer une structure,"
2120 :      PRINT"2 - Creer une fiche,"
2125 :      PRINT"3 - Visualiser l'ensemble des fiches,"
2130 :      PRINT"4 - Modifier une fiche,"
```

```
2140 :      PRINT"5 - Supprimer une fiche,"
2150 :      PRINT"6 - Lire sur disquette,"
2160 :      PRINT"7 - Sauvegarder sur disquette,"
2165 :      PRINT"8 - Sortir du Programme."
2170 :      LOCATE 0,19
2180 :      PRINT"Votre choix (1..8) ?"
2190 :      B$=INPUT$(1) : B=ASC(B$)-48
2200 :
2210 :      IF B<1 OR B>8 THEN 2000 'Reponse non valide
2220 :      ON B GOSUB 3000,3500,4000,4500,5000,5500,
6000,6500
2230 :
2240 :      IF FIN=1 THEN 2000 'Boucle Principale
2250 :
2260 : RETURN
2270 : REM ****************************************
3000 : REM Creation de structure
3010 :
3020 :   REM Titre
3030 :
3040 :      CLS : COLOR 1
3050 :      PRINT"      CREATION DE STRUCTURE"
3060 :      COLOR 6 : LOCATE 0,4
3070 :
3080 :   REM Creation
3090 :
3100 :      IF NA=0 THEN 3150
3110 :      PRINT"Structure deja existante."
3120 :      INPUT"On continue (O/N) ";R$
3130 :      IF R$="N" THEN 3240
3140 :
3150 :      INPUT"Nombre d'articles par fiche ";NA : PR
3160 :                                              INT
3170 :      FOR I=1 TO NA
3180 :        PRINT"Libelle article"I; : INPUT LA$(I)
3190 :      NEXT I
3200 :
3210 :      PRINT"Structure creee."
3220 :      GOSUB 9000 'Message de fin de creation
3230 :
3240 : RETURN
3250 : REM ****************************************
3500 : REM Creation de fiche
3510 :
3520 :   REM Titre
3530 :
3540 :      CLS : COLOR 1
3550 :      PRINT"     CREATION DE FICHE"
```

```
3560 :      COLOR 6 : LOCATE 0,4
3570 :
3580 :    REM Creation
3590 :
3600 :      IF NA=0 THEN PRINT"Structure absente." : GO
TO 3690
3610 :
3620 :      INPUT"No d'identification ";NI
3625 :      IF A$(1,NI)<>"" THEN PRINT"Fiche existante.
" : GOTO 3690
3630 :
3635 :      PRINT
3640 :      FOR I=1 TO NA
3650 :        PRINT LA$(I); : INPUT A$(I,NI)
3660 :      NEXT I
3670 :
3680 :      PRINT"Fiche en memoire."
3685 :      IF NI>NF THEN NF=NI
3690 :      GOSUB 9000 'Messa9e de fin de creation
3700 :
3710 : RETURN
3720 : REM ****************************************
4000 : REM Visualisation des fiches
4010 :
4020 :    REM Visualisation
4030 :
4040 :      FOR I=1 TO NF
4050 :        CLS : PRINT"Fiche"I : PRINT
4060 :        FOR J=1 TO NA
4070 :          PRINTLA$(J)"   : "A$(J,I)
4080 :        NEXT J
4090 :        GOSUB 9000 'Messa9e intermediaire
4100 :      NEXT I
4110 :
4120 :      LOCATE 0,20 : PRINT"Toutes les fiches ont e
te visualisees."
4140 :      GOSUB 9000 'Messa9e de fin de visualisation
4150 :
4160 : RETURN
4170 : REM ****************************************
4500 : REM Modification d'une fiche
4510 :
4520 :    REM Titre
4530 :
4540 :      CLS : COLOR 1
4550 :      PRINT"MODIFICATION D'UNE FICHE"
4560 :      COLOR 6 : LOCATE 0,4
4570 :
```

```
4580 :    REM Modification
4590 :
4600 :      INPUT"No d'identification ";NI
4610 :
4620 :      IF A$(1,NI)<>"" THEN GOTO 4650
4630 :      PRINT"Cette fiche n'est pas en memoire." :
GOTO 4780
4640 :
4650 :      REM Affichage de la fiche
4660 :
4670 :        FOR I=1 TO NA
4680 :          PRINT LA$(I)"  : ";A$(I,NI)
4690 :        NEXT I
4700 :
4710 :      REM Saisie de la modification
4720 :
4725 :        LOCATE 0,5
4730 :        FOR I=1 TO NA
4740 :          PRINT LA$(I)"  "; : INPUT A$(I,NI)
4750 :        NEXT I
4760 :
4770 :      PRINT"Modification effectuee."
4780 :      GOSUB 9000 'Message de fin de modification
4790 :
4800 : RETURN
4810 : REM ****************************************
5000 : REM Suppression d'une fiche
5010 :
5020 :   REM Titre
5030 :
5040 :      CLS : COLOR 1
5050 :      PRINT"SUPPRESSION D'UNE FICHE"
5060 :      COLOR 6 : LOCATE 0,4
5070 :
5080 :   REM Suppression
5090 :
5100 :      INPUT"No d'identification ";NI
5110 :
5120 :      IF A$(1,NI)<>"" THEN 5150
5130 :      PRINT"Cette fiche n'est pas en memoire."
5140 :
5150 :      FOR I=1 TO NA
5160 :        PRINT LA$(I)"  : "A$(I,NI)
5170 :      NEXT I
5180 :
5190 :      PRINT:INPUT"Etes-vous sur (O/N) ";R$
5200 :      IF R$="N" THEN 5300
5210 :
```

```
5220 :      FOR I=1 TO NA
5230 :        A$(I,NI)=""
5240 :      NEXT I
5250 :
5255 :      PRINT"Fiche supprimee." : NF=NF-1
5259 :
5260 :      REM Suppression physique
5261 :
5262 :        FOR I=NI TO NF
5263 :          FOR J=1 TO NA
5264 :            A$(J,I)=A$(J,I+1)
5265 :          NEXT J
5266 :        NEXT I
5267 :        FOR J=1 TO NA
5268 :          A$(J,I)=""
5269 :        NEXT J
5270 :
5280 :      GOSUB 9000 'Message de fin de suppression
5290 :
5300 : RETURN
5310 : REM ****************************************
5500 : REM Lecture disquette
5510 :
5520 :   REM Titre
5530 :
5540 :      CLS : COLOR 1
5550 :      PRINT"       LECTURE DISQUETTE"
5560 :      COLOR 6 : LOCATE 0,4
5570 :
5580 :   REM Lecture
5590 :
5600 :      PRINT"Cette commande detruira toute fiche"
5610 :      PRINT"ou structure en memoire."
5620 :      PRINT:INPUT"On continue (O/N) ";R$
5630 :      IF R$="N" THEN 5840
5640 :
5650 :      INPUT"Nom du fichier ";NF$
5660 :      OPEN"I",#1,NF$
5670 :
5680 :      INPUT #1,NF,NA
5690 :
5700 :      FOR I=1 TO NA
5710 :        INPUT #1,LA$(I)
5720 :      NEXT I
5730 :
5740 :      FOR I=1 TO NF
5750 :        FOR J=1 TO NA
5760 :          INPUT #1,A$(J,I)
```

```
5770 :        NEXT J
5780 :      NEXT I
5790 :
5800 :      CLOSE #1
5810 :
5820 :      PRINT"Fichier en memoire."
5830 :
5840 :      GOSUB 9000 'Messa9e de fin de lecture
5850 :
5860 : RETURN
5870 : REM ******************************************
6000 : REM Ecriture dis9uette
6010 :
6020 :    REM Titre
6030 :
6040 :      CLS : COLOR 1
6050 :      PRINT"    ECRITURE DISQUETTE"
6060 :      COLOR 6 : LOCATE 0,4
6070 :
6080 :    REM Ecriture
6090 :
6100 :      INPUT"Nom du fichier ";NF$
6110 :
6120 :      OPEN"O",#1,NF$
6130 :
6140 :      WRITE #1,NF,NA
6150 :
6160 :      FOR I=1 TO NA
6170 :        WRITE #1,LA$(I)
6180 :      NEXT I
6190 :
6200 :      FOR I=1 TO NF
6210 :        FOR J=1 TO NA
6220 :          WRITE #1,A$(J,I)
6230 :        NEXT J
6240 :      NEXT I
6250 :
6260 :      CLOSE #1
6270 :      PRINT"Fichier sauve9arde."
6280 :
6290 :      GOSUB 9000 'Messa9e de fin de sauve9arde
6300 :
6310 : RETURN
6320 : REM ******************************************
6500 : REM Fin du Pro9ramme
6510 :
6520 :    REM Titre
6530 :
```

```
6540 :      CLS : COLOR 1
6550 :      PRINT"        FIN DU PROGRAMME"
6560 :      COLOR 6 : LOCATE 0,4
6570 :
6580 :    REM Fin
6590 :
6600 :      PRINT"Cette commande detruira toute fiche"
6610 :      PRINT"ou structure en memoire."
6620 :      PRINT:INPUT"On continue (O/N) ";R$
6630 :      IF R$="O" THEN FIN=-1 ELSE FIN=1
6640 :
6650 : RETURN
6660 : REM ****************************************
9000 : REM Message de fin d'action
9010 :
9020 :    COLOR 3 : PRINT
9030 :    PRINT"Pressez une touche pour continuer"
9040 :    LOCATE 1,1,0 'Suppression du curseur
9050 :    COLOR 6
9060 :    C$=INPUT$(1) 'Attente de l'appui
9070 :
9080 : RETURN
```

Hard-Copy d'écran basse résolution

PRESENTATION

Ce petit utilitaire permet de faire une copie d'écran basse résolution sur une imprimante 80 colonnes. Il doit être inséré dans le programme qui nécessite la copie d'écran et est appelé par "GOSUB 10000".

Pour que la copie soit activée, tapez sur une touche quelconque du clavier.

ORGANIGRAMME DE L'UTILITAIRE

Le programme possède la structure linéaire définie en annexe.

IDENTIFICATION DES VARIABLES

I : index de ligne.

J : index de colonne.

```
10000 : REM HARD COPY D'ECRAN BASSE RESOLUTION
10010 :
10020 :   A$=INPUT$(1) 'Attente d'une action clavier
10030 :
10040 :   OPEN"O",#3,"LPRT:(80)"
10050 :   FOR I=0 TO 24
10060 :     FOR J=0 TO 39
10070 :       PRINT #3,CHR$(SCREEN(J,I));
10080 :     NEXT J
10090 :     PRINT #3
10100 :   NEXT I
10110 :   CLOSE #3
10120 :
10130 : RETURN
```

Annexe

Cette annexe est similaire à celle du tome 1. Elle décrit le type de programmation employé dans ce manuel.

"Qu'est-ce qu'un programme ?"

Voici une réponse qui ne se veut pas exhaustive.

C'est un ensemble d'instructions ou mots-clés, stockés en mémoire, qui ont pour but d'activer les ressources matérielles de l'ordinateur.

Si ces instructions sont mises bout à bout, le programme risque d'être incompréhensible. Il est alors comparable à un texte en prose excluant ponctuation, paragraphes et chapitres. Cela n'est pas très grave si le programme est simple. Si le programme est complexe, il devient très vite impossible à déchiffrer.

Il apparaît donc que, comme en littérature, un programme manipulant un grand nombre de concepts doit comporter ponctuation, paragraphes et chapitres.

En informatique, la ponctuation porte le nom de structure, les paragraphes sont appelés modules et les chapitres niveaux.

Comme en littérature, un programme comporte :

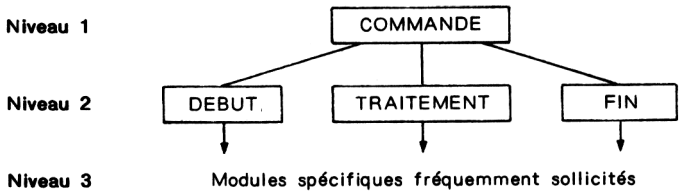
- plusieurs niveaux ;
- chaque niveau contient plusieurs modules ;
- à l'intérieur d'un module, la structure sépare et organise les concepts manipulés.

Dans ce livre, les programmes sont relativement simples. Ils comportent une structure à deux ou trois niveaux :

- le **niveau 1** comporte un seul module appelé **module de commande** ;

- le **niveau 2** se décompose en trois modules appelés : **début, traitement et fin** ;
- le **niveau 3**, s'il existe, se décompose en divers modules réalisant des tâches simples, fréquemment sollicitées.

Le schéma suivant ou "graphe hiérarchique" détaille cette structure.



Le module "**DEBUT**" réalise la présentation et l'initialisation des variables et constantes du programme.

Le module "**TRAITEMENT**" constitue le corps du programme. C'est lui qui contient la plupart des concepts manipulés pour réaliser le programme.

Le module "**FIN**" présente le résultat final, les remarques ou commentaires, et déconnecte l'ordinateur du programme lancé par la commande "RUN".

Les programmes les plus simples ne nécessitent pas de structure hiérarchique à deux ou trois niveaux. Ils possèdent seulement un niveau et leur structure est dite linéaire...

En conclusion, plus un programme est compliqué, plus il est nécessaire qu'il soit structuré et possède des niveaux différents.

Dans la pratique, tout programmeur aura intérêt d'écrire un programme de la sorte, car cela facilite grandement la lisibilité (et donc le dépistage des erreurs) et la maintenance.

Imprimé en Belgique
sur les presses du Scorpion
D/1984/3548/11
N° d'édition 87109.016.1
I.S.B.N. 2.87109.016.5

MegaO POCHE

BOITE A OUTILS POUR MO5 et TO7/70 18 PROGRAMMES EN BASIC

Qui n'a jamais regretté de ne pas avoir sous la main le tournevis ou la pince pour faire tel ou tel bricolage? Les ordinateurs par leur aspect universel sont comme les boîtes à outils (et les auberges espagnoles...): on y trouve ce que l'on y met. Cette collection MegaO-Poche n'a pas la prétention de livrer le secret des grandes applications informatiques, mais bien de mettre à la disposition des utilisateurs des petits programmes tout prêts qui leur permettront de résoudre de nombreux problèmes de la vie quotidienne.

35,00 FF

ISBN 2.87109.016.5

DIFFUSE PAR PSI

Couverture création et dessin Françoise Guillot exécution Agraph